

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Інститут телекомунікаційних систем

Кафедра Інформаційно-телекомунікаційних мереж

«На правах рукопису»

УДК _____

«До захисту допущено»

Завідувач кафедри

_____ Лариса ГЛОБА

«__» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інформаційно-комунікаційні
технології»**

зі спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: «Програмне забезпечення для проведення незалежного on-line
тестування з використанням qr кодів»**

Виконав:

студент VI курсу, групи ТІ-91мп

Гієнко Юрій Олександровч _____

Керівник:

доцент кафедри ІТМ ІТС, доцент, к.т.н.

Правило Валерій Володимирович _____

Рецензент:

доцент кафедри ТК ІТС, доцент, к.н.т.

Явіся Валерій Сергійович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Інформаційно-телекомунікаційних мереж

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інформаційно-комунікаційні технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Лариса ГЛОБА

« ____ » _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Гієнку Юрію Олександровичу

1. Тема дисертації «Програмне забезпечення для проведення незалежного on-line тестування з використанням qr кодів», науковий керівник дисертації доцент кафедри інформаційно-телекомунікаційних мереж ІТС Правило Валерій Володимирович, доцент, к.т.н., затверджені наказом по університету від «03» листопада 2020 р. № 3208-с
2. Термін подання студентом дисертації 10.12.2020 р.
3. Об'єкт дослідження:
4. Предмет дослідження:
5. Перелік завдань, які потрібно розробити:
 - 5.1. Постановка проблеми та оцінка шляхів вирішення.
 - 5.2. Планування гнучкої платформи для тестування з використанням QR кодів.
 - 5.3. Проведення дослідження , спираючись на дані ,отриманні в ході опитування.

5.4. Реалізувати web додаток на базі Angular 8 та провести тестування основних модулів.

6. Орієнтовний перелік ілюстративного матеріалу:

1. Тема, актуальність, мета, задачі.
2. Оцінка підготовленості учнів користуватися технологією QR .
3. Огляд здатності дітей до сприйняття матеріалу , шляхи стимулювання наступних заохоченнями та аналіз сучасних технологій обробки даних.
4. Підхід до реалізації онлайн додатку та створення власних QR кодів.
5. Розробка алгоритмів вибору завдань до тесту та видалення QR коду після закінчення часу тестування.

7. Орієнтовний перелік публікацій:

1. Перспективи телекомунікацій 2019 – Прискорення алгоритмів шифрування реалізованих на основі графічних процесорів, Пилипчук А. О.Правило В.В.
2. Перспективи розвитку інформаційно-телекомунікаційних технологій та систем 2019 – Використання технології Cuda для прискорення алгоритму шифрування.
3. Перспективи телекомунікацій 2020 – Моделі ідентифікації та аутентифікації в системах хмарних вичислень для веб-серверів та мобільних додатків з інтелектуальною підтримкою вибору, Пилипчук А. О.

9. Дата видачі завдання 10.09.2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення з технологією QR кодів	10.09.2019-10.10.2019	виконано
2	Огляд відомих рішень з впровадження технології QR	11.10.2019-11.11.2019	виконано
3	Розробка алгоритмів створення унікальних тестів та підключення бази даних MySql для зберігання інформації.	12.11.2019-12.01.2020	виконано
4	Впровадження основних сутностей проекту та створення архітектури проекту	13.01.2020-13.06.2020	виконано
5	Створення Rest API для доступу до бази даних.	13.06.2019-13.08.2020	виконано
6	Реалізація програмного забезпечення.	14.08.2020-14.11.2020	виконано

Студент

Юрій ГІСНКО

Керівник роботи

Валерій ПРАВИЛО

РЕФЕРАТ

Робота містить 91 сторінок, 19 рисунків та 8 таблиць. Було використано 20 джерела.

Мета роботи: розробка web платформи для проведення ,створення та запровадження незалежного тестування . Розробити алгоритм ідентифікації користувача , шляхом генерації унікальних QR кодів та CRUD операціями , пов'язаними з ними , що необхідні для коректної роботи платформи. Вирішення ефективності, масштабованості та об'єму даних платформи та доцільність зберігання їх в базі даних . Розробка бази даних тестів з низки звичних та спеціальних дисциплін , що адаптуються під рівень знань конкретного учня , шляхом обробки його відповідей на питання різного рівня складності . Створити глобальну базу учбових матеріалів , тестів та завдань , що значно полегшить роботу вчителів та дасть змогу контролювати самостійну роботу учнів над матеріалами поза школою . Надати права обмежених адміністраторів для викладачів , тим самим дати змогу створювати спеціальні тести для оцінки залишкових знань учнів . Розробка алгоритму підбору унікальних завдань до тесту , аби виключити можливість колективного вирішення тестів та списування . Алгоритм видалення QR кодів після завершення тестування учнями. Створення системи розподілення балів за різними рівнями складності тестів та активністю користувачів .

Ключові слова: алгоритм, децентралізовані тести , QR код, обмежені права адміністратора , рівень знань , глобальна база даних , метод визначення , спеціальні кімнати.

ABSTRACT

The work contains 91 pages, 19 figures and 8 tables. 20 sources have been used.

Goal: development of a web platform for conducting, creating and implementing independent testing. Develop a user identification algorithm, by generating unique QR codes and CRUD operations related to them, which are necessary for the correct operation of the platform. Decide on the efficiency, scalability and volume of platform data and the feasibility of storing them in a database. Development of a database of tests in a number of familiar and special disciplines that adapt to the level of knowledge of a particular student, by processing his answers to questions of varying difficulty disciplines they intend to study in the future. According to the selected "fraction", the user will have access to a faction chat, where he will be able to communicate with other users and discuss the urgent tasks and receive assistance from the faction.

Create a global database of teaching materials, tests and tasks that will greatly facilitate the work of teachers and allow students to control the independent work of materials on materials outside school. Grant the rights of restricted administrators to teachers, thus enabling the creation of special tests to assess students' residual knowledge. Development of an algorithm for selecting unique tasks for the test to eliminate the possibility of collective solving of tests and writing off. Algorithm for deleting QR codes after testing by students. Creating a system of distribution of points for different levels of test complexity and user activity.. Creating a system of distribution of points for different levels of test complexity and user activity.

Key words: algorithm, decentralized tests, limited administrator privileges, QR codes, knowledge level, global database, determination method, special rooms.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1	12
ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ СЕРЕД ШКОЛЯРІВ ТА ЇХ БАТЬКІВ .	
ОБРОБКА РЕЗУЛЬТАТІВ , ОТРИМАНИХ ПІД ЧАС ДОСЛІДЖЕННЯ ТА	
НАПИСАННЯ ВИСНОВКУ ПО ВИКОНАНІЙ РОБОТІ	12
1.1 Вступ.....	12
1.2 Етапи дослідження та оприлюднення результатів	13
Висновки	18
РОЗДІЛ 2	19
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ОПИС ТЕХНОЛОГІЙ ,	
ЩО ВИКОРИСТОВУВАЛИСЯ ДЛЯ СТВОРЕННЯ ОНЛАЙН ПРОЕКТУ ..	19
2.1. Трирівнева архітектура.....	19
2.2 Основна інформація про «Коди швидкої відповіді»	25
2.2 Огляд архітектури Angular 8	40
2.3 Модулі	41
2.4 Компоненти.....	42
2.5 Шаблони, директиви та прив'язка даних	43
2.6 Послуги та ін'єкції залежності	44
2.7 Маршрутизація	44
2.8 Введення в модулі	46
2.9 Метадані NgModule.....	46
2.10 NgМодулі та компоненти	47
2.11 Модулі NgModules і JavaScript	49
2.12 Angular libraries.....	49
2.13 Dotnet core	50
2.14 Інструменти для роботи з .NET Core	52
2.15 Існуючі рішення	53
2.16 Платформа корпоративного рівня для всіх	53

2.17 Azure	55
Висновки	72
РОЗДІЛ 3	74
ОГЛЯД АНАЛОГІВ ТА ПОРНІВНЯННЯ ІЗ ДОДАТКОМ	74
3.1 Аналіз конурентів.....	74
Висновки	78
4.1. Опис ідеї проекту	80
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	90
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	91

ПЕРЕЛІК СКОРОЧЕНЬ

БД

база даних

ПЗ

програмне забезпечення

ВСТУП

Актуальність: онлайн тестування стало невід’ємною частиною учбового процесу в багатьох учбових закладах. Та із зростанням попиту зростають і потреби користувачів до платформ онлайн тестування. Досить вагомою проблемою є списування та можливість проходження тесту декілька разів. При правильно спланованій платформі можна отримати максимально незалежний та об’єктивний результат .

Більшість учнів навчилися використовувати допомогу один-одного при співпадині питань у білеті. В подальшому житті це може призвести до деградації поколінь, жахлива перспектива , оскільки при використуванні стороньої допомоги вірогідність незасвоєння матеріалу дуже висока. Суспільство, на мою думку, потребує раціонального рішення, тому що загальна необізнаність в елементарних питаннях спричиняє серйозніші проблеми у дорослому житті.

На мою думку використання QR технології та правильно спланованій платформі з використанням останніх технологій допоможе створити якнайкращі умови для проходження максимально незалежного тестування. В разі ігнорування списування можна отримати дуже невтішні наслідки у вигляді невпевнених у своїх силах учнях , які будуть шукати лише допомоги , а не вирішувати питання самостійно.

Мета роботи: розробка онлайн додатку з максимально прозорим та ефективним алгоритмом незалежного тестування за допомогою технології QR . Провести опитування серед батьків та школярів на базі середніх загальноосвітніх шкіл Києва . На основі отриманих даних створити алгоритми оцінки . Розробка бази даних тестів з низки звичних та спеціальних дисциплін , що адаптуються під рівень знань конкретного учня , шляхом обробки його відповідей на питання різного рівня складності . Створити глобальну базу учбових матеріалів , тестів та завдань , що значно полегшить роботу вчителів та

дасть змогу контролювати самостійну роботу учнів над матеріалами поза школою

Для досягнення мети дослідження було поставлено та вирішено такі **Основні задачі** : розробка онлайн платформи із двофакторною аутентифікацією за допомогою QR коду

Теоретичний результат дослідження: надання можливості створення , впровадження та проходження онлайн тестування за умов , максимально наближених до незалежних.

Практичний результат роботи: створення онлайн додатку та інтеграція в глобальну мережу інтернет. Розробка front-end та back-end складових онлайн додатку із використанням бібліотеки кодів швидкої відповіді.

РОЗДІЛ 1

ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ СЕРЕД ШКОЛЯРІВ ТА ЇХ БАТЬКІВ . ОБРОБКА РЕЗУЛЬТАТІВ , ОТРИМАНИХ ПІД ЧАС ДОСЛІДЖЕННЯ ТА НАПИСАННЯ ВИСНОВКУ ПО ВИКОНАНІЙ РОБОТІ

1.1 Вступ

Освіта – це цілеспрямована праця над собою, щоденна та тяжка, засвоєння знань та умінь. Прикладаючи свої зусилля, кожен день виконуючи які-небудь завдання можна досягти не абиякого розвитку, заради кращого майбутнього для нас та наших нащадків. Але ж, чесно кажучи, не всім це вдається через лінь або що. Тому, кращим рішенням для невеликої перемоги над собою є стимулювання.

Під час навчання в онлайн режимі викладач не може абсолютно контролювати учня на відстані із зрозумілих причин. Саме це і призводить до списування та можливості використовувати чужі відповіді у своєму тесті.

В сучасному світі вимоги до освіти інші, вибору, свободи та технологій більше. Науково-технічний прогрес призвів до того, що доступ до знань та інформації занадто легкий. В такий час ера можливості стає занадто близька до ери ліні. Користуватися пристроями задля пошуку інформації можуть навіть малі діти, що водночас є небезпечним, адже наразі дуже багато невірної чи навіть шкідливої інформації для малечі в інтернеті. В подальшому житті це може призвести до деградації поколінь, жахлива перспектива. Суспільство, на мою думку, потребує раціонального рішення, тому що загальна необізнаність в елементарних питаннях спричиняє серйозніші проблеми у дорослому житті. Чи можливо це змінити не зупинивши розвиток людства?

Так, звісно, це просто необхідно! Потрібно використовувати доречні методи стимулювання до навчання. Наприклад, для дітей це може бути заохочення від батьків у вигляді кишенькових грошей/смачних ласощів/прогулянки за виконане домашнє завдання. Але ж ми у 21 столітті, часі

– коли майже все діджиталізовано. Тому пропоную додаток, який не тільки буде дуже зручним, 100% чесним і справедливим до кожної дитини, а й розвиватиме в ній самостійність, поступово привчаючи її до незалежності від батьків. Все це відбуватиметься прямісінько в смартфоні, але через власне навчання, шляхом заохочення цікавими призами, бонусами тощо.

1.2 Етапи дослідження та оприлюднення результатів

Для розуміння чи готові учні та їх батьки до сприйняття нового методу оцінки було проведено опитування та обробка відповідей:

1. Серед батьків – про рівень розумової активності та відповідального ставлення дітей до навчання (200 батьків учнів середніх класів);
2. Серед учнів загальноосвітніх навчальних закладів – про зацікавленість у навчанні та їх баченні щодо впровадження QR технології (100 учнів середніх класів);
3. Серед студентів коледжів та вищих навчальних закладів – про зацікавленість у навчанні та їх баченні щодо QR технології (200 студентів коледжів та 300 з ВНЗ);
4. Серед вчителів та викладачів – про їх думку щодо алгоритму створення та компанування тестів (135 вчителів та 165 викладачів);
5. Тестування продукту – робота в додатку, його зручність (5 учнів середніх класів);

Таблиця 1.1

Опитування батьків

Питання	Позитивні відповіді	Негативні відповіді
Чи вважаєте ви, що ваша дитина відповідально ставиться до навчання ?	28	172
Ви вважаєте, що ваша дитина вчиться на достатньому рівні?	69	131

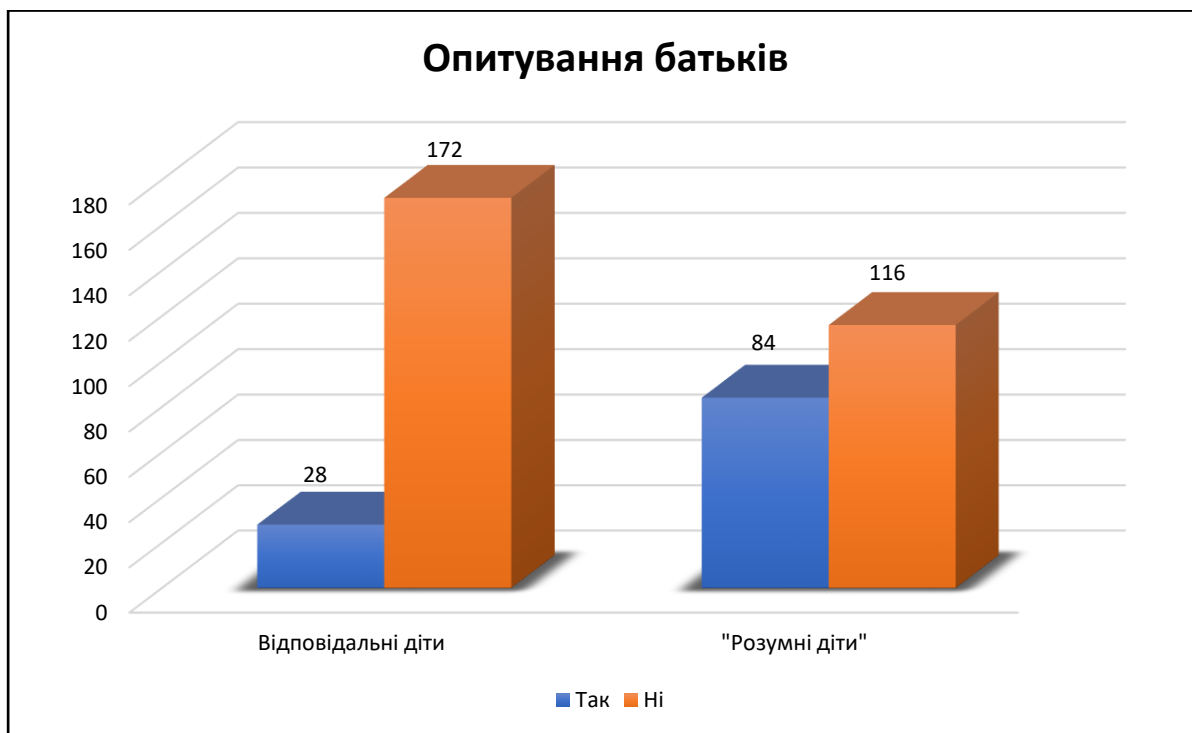


Рис 1.1 Опитування батьків

Виходячи з цих даних, ми бачимо, що, дивлячись правді в очі, батьки розуміють реальне становище на сьогоднішній день. Той рівень знань, якими володіють їх діти – не задовільняє вимоги батьків та теперішнього суспільства в цілому, але розумних дітей достатньо. Також, як не прикро це визнавати, відповідальне ставлення учнів до навчання дійсно в поганому стані. Якщо не має самого бажання, той не буде результату, прогресу. Це потрібно починати розвивати в дітях з дитинства і потрошку підкріплювати заохоченнями за певні гарні результати, головне без примусу, як це було років 40-50 тому з нашими батьками та дідусями. Так, такі методи дієві, але ж ціною індивідуальності та свободи дитини.

Таблиця 1.2

Опитування школярів

	Позитивні відповіді	Негативні відповіді
Чи маєте ви жагу до навчання?	22	78

Чи вмієте ви користуватись сканером	QR	83	17
-------------------------------------	----	----	----



Рис 1.2 Опитування учнів

Це дослідження дало нам зрозуміти, що у дітей немає стимулу до навчання, такого, щоб вони самі цього хотіли, а не змушували батьки. Така ситуація завжди була присутня, але чи не настав час це змінити, покращити якість життя дітей та отримати гарний результат у майбутньому? Як ми бачимо, бажання мати щось, окрім оцінки, за виконані завдання дійсно велике і учні краще за всіх знають, що задовольнить їх. Це можуть бути: Гроші, їжа (солодощі), іграшки, ігрові пристрої, квитки у кіно, парк розваг та на цікаві івенти. Мені здається, це не велика ціна за щасливі обличчя дітлахів та їх розумні голови.

Таблиця 1.3.

Опитування студентів

Питання	Позитивні відповіді	Негативні відповіді	Пропозиції щодо нагород
Чи маєте ви жагу до навчання?	91	409	Стипендія, відпочинок,

Чи хотіли б ви отримувати за оцінки якусь нагороду та яку саме?	376	124	вихідні, знижки або сертифікати на розваги.
---	-----	-----	---



Рис 1.3 Опитування студентів

Отже, студенти, як більш самостійна верства, мають невеликий відсоток таких, що зовсім не зацікавлені ні в навчанні, ні в заохоченнях, тому сегмент молоді буде меншим, за школярів. Але, якщо зрівняти скільки людей не мають жаги проте мають бажання отримувати винагороду, то можна зробити висновок – надія не втрачена, ми зможемо відновити такий цінний трудовий ресурс. Студентство більш цікавлять грошові винагороди у вигляді підвищення стипендій, додаткових виплат, знижок на розваги та відпочинок.

Таблиця 1.4

Опитування вчителів та викладачів

Питання	Позитивні відповіді	Негативні відповіді
На вашу думку, система заохочень буде корисною?	235	65
Чи хотіли б ви, в подальшому, працювати в такій системі?	180	120

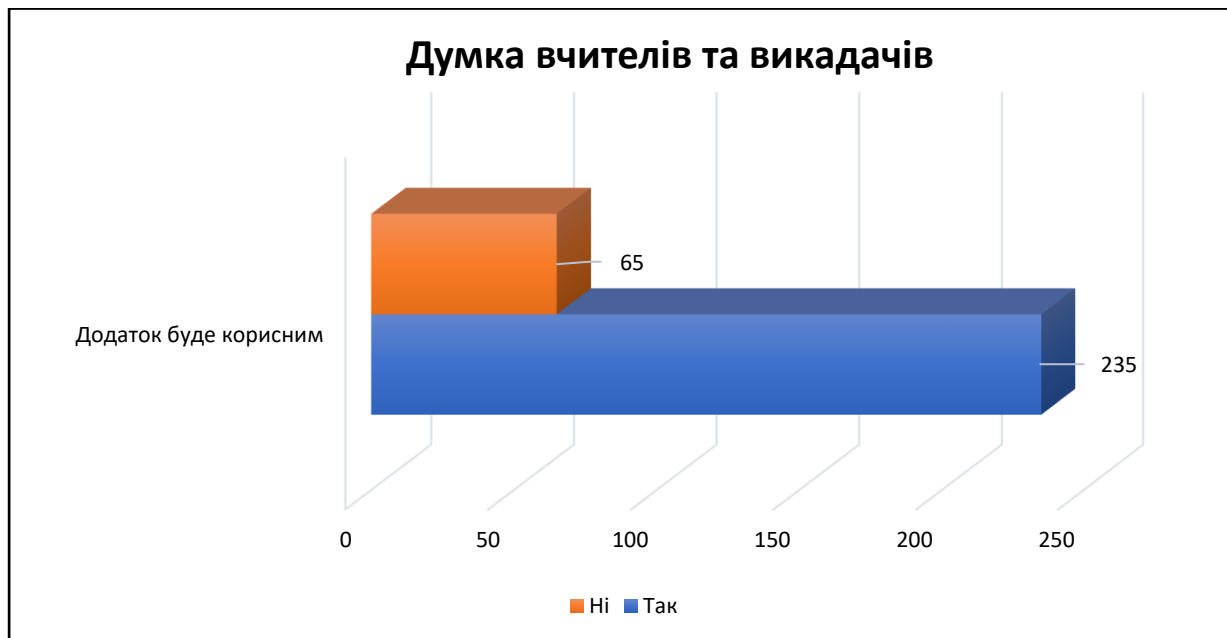


Рис. 1.4 Думка вчителів та викладачів



Рис. 1.5 Готовність до співпраці

Отож, на думку викладачів та вчителів, ідея додатку для заохочення дітей до навчання буде корисною. Серед них 72% хотіли б співпрацювати в подальшому з нашою програмою.

Тестування продукції.

Для тестування додатку було обрано 10 учнів з 5-9 класів. Вони виконали по одному завданню, оцінили інтерфейс, дизайн та складність використання програми, також прослухали консультаційну інформацію щодо того, як працює

система заохочень та отримали винагороду відповідно вирішеному завданню. Всі результати ми підбили та зібрали в одну таблицю (табл.4).

Таблиця 1.5.

Оцінка додатку

Критерій \ Оцінка	Погано	Задовільно	Добре	Відмінно
Дизайн	-	2	1	7
Зручність у використанні	1	1	1	7
Складність системи оцінювання	1	-	3	6
Варіанти винагород	-	3	2	5
Загальне враження	-	-	4	6

Висновки

В результаті проведення цих досліджень, з точністю можна сказати, що справа потребує рішення, але перед тим, більш глибокого дослідження задля уникнення ризиків при запуску додатку на ринок. Розумних дітей не мало, серед усіх, але бажання навчатися є лише в одиниць, потрібно це змінювати та впроваджувати системи щодо заохочень дітей до навчання та виконання завдань. Саме таким шляхом ми зможемо змінити ставлення молоді до освіти та досягти кращого майбутнього – розумного та вільного у своєму виборі. Основним напрямком роботи додатку буде створення умов для максимально незалежного тестування за допомогою QR технології.

РОЗДІЛ 2

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ОПИС ТЕХНОЛОГІЙ , ЩО ВИКОРИСТОВУВАЛИСЯ ДЛЯ СТВОРЕННЯ ОНЛАЙН ПРОЕКТУ

Платформа потребує досить чіткого планування . Програмне забезпечення грає високу у ролі у створенні та впровадженні даної платформи.

Платформу успішно розроблено за допомогою Angular 8 та розгорнуто на хмарному сховищі Azure . За допомогою пакетного менеджера було додані основні бібліотеки ,що використовувались в проєкті:Bootstrap 4, Angular Material, Font Awesome, DevExpress , а також імпортовано модуль для створення QR кодів.

Всі сутності та їх поля було прописано та розгорнуто разом із базою даних MySQL

Програму було розроблено дотримуючись трирівневої архітектури.

2.1. Трирівнева архітектура

Існує безліч різних видів і типів архітектур, які успішно застосовуються. Однієї з найбільш використовуваних є класична трирівнева система, яка передбачає поділ додатка на три рівні.

огляд архітектури

- Клієнт - це інтерфейсний (зазвичай графічний) компонент, який представляє перший рівень, власне додаток для кінцевого користувача. Перший рівень не повинен мати прямих зв'язків з базою даних (за вимогами безпеки), бути навантаженим основний бізнес-логікою (за вимогами масштабованості) і зберігати стан додатки (за вимогами надійності). На перший рівень може бути винесена і зазвичай виноситься найпростіша бізнес-логіка: інтерфейсавторизації, алгоритми шифрування, перевірка вводятьсязначень на допустимість і відповідність формату, нескладні

операції (сортування, угруповання, підрахунок значень) з даними, вже завантаженими на термінал.

- Сервер додатків розташовується на другому рівні. На другому рівні зосереджена велика частина бізнес-логіки. Поза його залишаються фрагменти, що експортуються на термінали (див.вище), а також занурені в третій рівень збережені процедури і тригери.
- Сервер бази даних забезпечує зберігання даних і виводиться на третій рівень. Зазвичай це стандартна реляційна або об'єктно-орієнтована СУБД. Якщо третій рівень являє собою базу даних разом з збереженими процедурами, тригерами та схемою, яка описує додаток в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що зв'язує клієнтські компоненти з прикладної логікою бази даних.

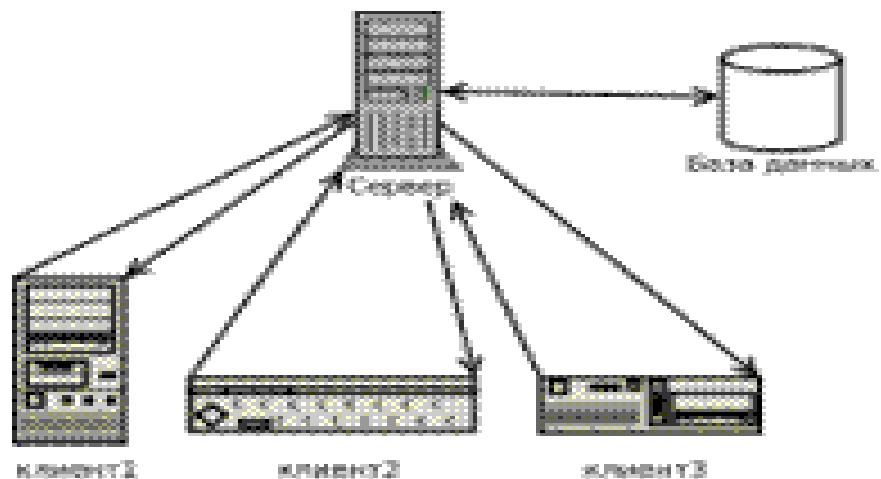


Рис. 2.1 трирівнева архітектура

У простій конфігурації фізично сервер додатків може бути поєднаний з сервером бази даних на одному комп'ютері, до якого по мережі підключається один або кілька терміналів. У «правильної» (з точки зору безпеки, надійності, масштабування) конфігурації сервер бази даних знаходиться на виділеному комп'ютері (або кластері), до якого по мережі підключені один або кілька серверів додатків, до яких, в свою чергу, по мережі підключаються термінали.

переваги У порівнянні з клієнт-серверної або файл-серверної архітектурою можна виділити наступні переваги трирівневої архітектури:

- Масштабованість
- Конфігурованість - ізольованість рівнів один від одного дозволяє (при правильному розгортанні архітектури) швидко і простими засобами переконфігурувати систему при виникненні збоїв або при плановому обслуговуванні на одному з рівнів
- Висока безпека
- Висока надійність
- Низькі вимоги до швидкості каналу (мережі) між терміналами і сервером додатків
- Низькі вимоги до продуктивності і технічним характеристикам терміналів, як наслідок зниження їх вартості. Терміналом може виступати не тільки комп'ютер, але і, наприклад, мобільний телефон.

Недоліки:

Недоліки впливають з достоїнств. У порівнянні з клієнт-серверної або файл-серверної архітектурою можна виділити наступні недоліки трирівневої архітектури:

- Більш висока складність створення додатків;
- Складніше в розгортанні і адмініструванні;
- Високі вимоги до продуктивності серверів додатків і сервера бази даних, а, значить, і висока вартість серверного обладнання;
- Високі вимоги до швидкості каналу (мережі) між сервером бази даних і серверами додатків.

Тут відразу треба сказати, що багаторівневої архітектурою часто позначають два не зовсім пов'язаних поняття: n-layer і n-tier. I layer, і tier, як правило, позначаються словом "рівень", іноді по відношенню до "layer" ще вживається слово "шар". Однак в обох випадках рівні будуть різного порядку.

Tier представляє фізичний рівень. Тобто якщо ми говоримо про трирівневу архітектуру, то n-tier додаток могло бути розділене на такі рівні: сервер бази даних, веб-додаток на веб-сервері і браузер користувача. Тобто кожен рівень представляв би особливий окремий фізичний процес, навіть якщо б і сервер баз даних, і веб-сервер, і браузер користувача знаходилися б на одному комп'ютері. Якби в якості клієнта альтернативно використовувалося мобільний додаток, то це був би ще один фізичний рівень.

Layer представляє логічний рівень. Тобто у нас може бути рівень доступу до даних, рівень бізнес-логіки, рівень уявлення, рівень сервісів і так далі. При цьому логічні рівні не збігаються з фізичними. Так, звичайно рівень надання в додатку ASP.NET містить і контролери, які обробляють введення, і уявлення, які відображаються в веб-браузері, тобто розділяється на два фізичних рівня. В даному випадку ми будемо говорити саме про логічних рівнях, тобто про n-layer архітектурі. Класична трирівнева система складається з наступних рівнів:

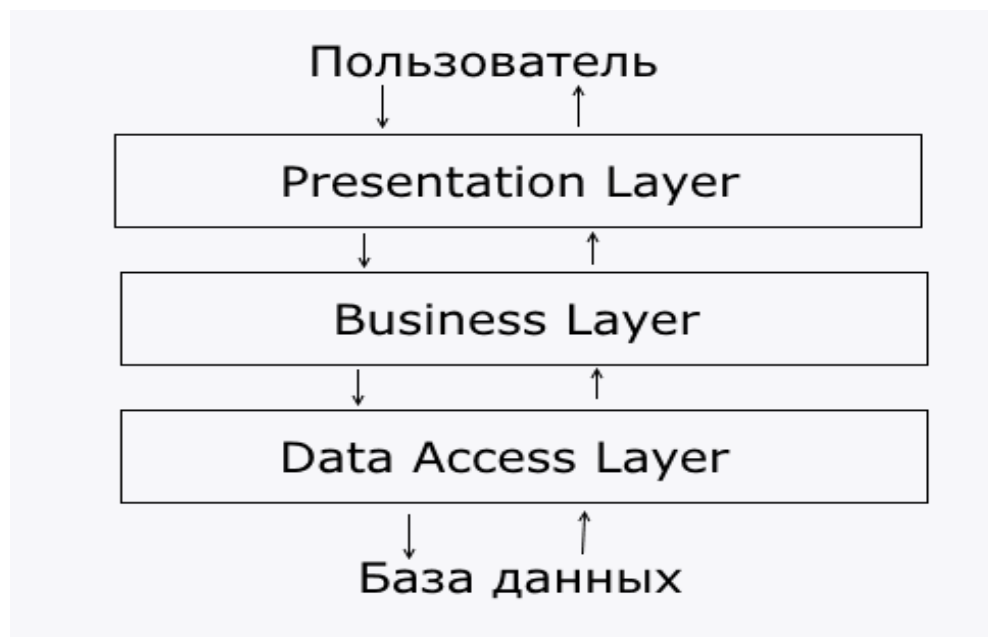


Рис. 2.2 Логічне представлення трирівневої архітектури

Presentation layer (рівень представлення): це той рівень, з яким безпосередньо взаємодіє користувач. Цей рівень включає компоненти для користувача інтерфейсу, механізм отримання введення від користувача. Стосовно до asp.net mvc на даному рівні розташовані уявлення і всі ті

компоненти, який складають призначений для користувача інтерфейс (стилі, статичні сторінки html, javascript), а також моделі уявлень, контролери, об'єкти контексту запиту.

Business layer (рівень бізнес-логіки): містить набір компонентів, які відповідають за обробку отриманих від рівня уявлень даних, реалізує всю необхідну логіку додатка, все обчислення, взаємодіє з базою даних і передає рівнем подання результат обробки.

Data Access layer (рівень доступу до даних): зберігає моделі, що описують використовувані суті, також тут розміщуються специфічні класи для роботи з різними технологіями доступу до даних, наприклад, клас контексту даних Entity Framework. Тут також зберігаються репозиторії, через які рівень бізнес-логіки взаємодіє з базою даних.

При цьому треба зазначити, що крайні рівні не можуть взаємодіяти між собою, тобто рівень представлення (стосовно ASP.NET MVC, контролери) не можуть безпосередньо звертатися до бази даних і навіть до рівня доступу до даних, а тільки через рівень бізнес-логіки .

Рівень доступу до даних не залежить від інших рівнів, рівень бізнес-логіки залежить від рівня доступу до даних, а рівень представлення - від рівня бізнес-логіки.

Компоненти, як правило, повинні бути слабо зв'язані (loose coupling), тому невід'ємною ланкою багаторівневих додатків є впровадження залежностей.

При чому про ASP.NET MVC ми говоримо перш за все стосовно до рівня уявлення, інші ж рівні можуть бути реалізовані незалежно і можуть використовуватися в додатках на інших технологіях, як Windows Forms, WPF і т.д. І, як правило, все додаток в цілому буде представляти рішення (solution) в Visual Studio, а окремі рівні - проекти. У той же час невірно думати, що якщо рівень обов'язково повинен відповідати окремий проект. При необхідності ми

можемо роздібнити один рівень на кілька проектів, головне, щоб його функціонал представляв єдину логічну ланку.

Існує величезна кількість технологій та програмного забезпечення для створення та підтримки онлайн додатків . Всі підходи можна поділити на 3 великі групи :

- Підходи ,що базуються на програмуванні або скриптах : зовнішні програми або скрипти.
- Підходи , що базуються на використанні шаблонів web-сторінок.
- Підходи,що базуються на використанні об'єктних середовищ (каркаси,фреймворки).

При створенні «Brain Benefits» було використано програмний підхід , оскільки при підготуванні плану проекту , стало зрозуміло , що додаток матиме значну частку специфічного функціоналу , що базується на самописних функціях та скриптах , що взаємодіють з масивами інформації , що зберігається в базі даних MySQL.

Проект розроблено дотримуючись архітектурного шаблону MVC . Відповідно до патерну MVC весь додаток ділиться на три логічних компонента: Модель (Model), Представлення (View) і Контролер (Controller). Логіка роботи web-додатки з використанням архітектури MVC показана на рис. 2.3.

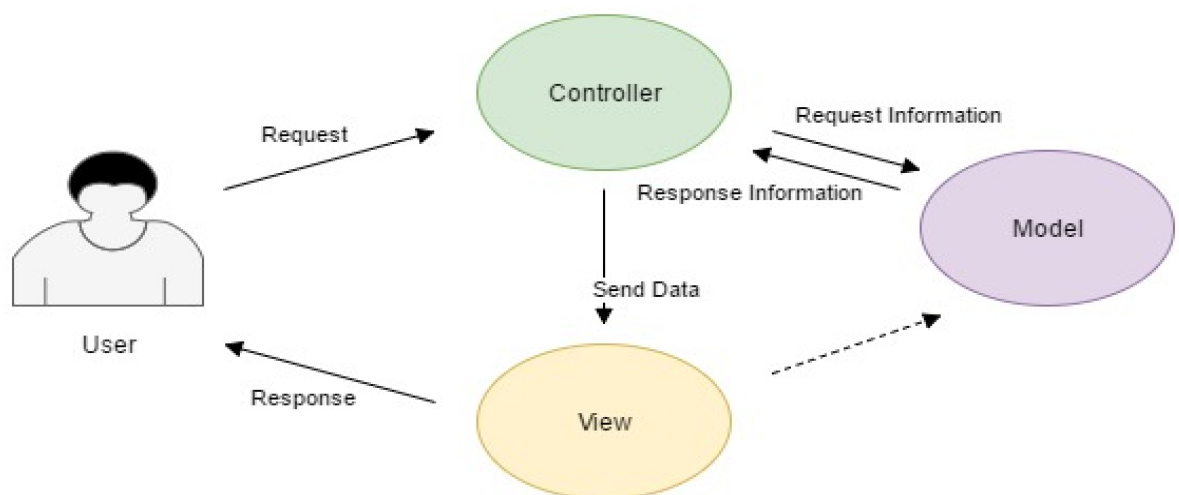


Рис. 2.3 Модель MVC

- Модель (Model) - це набір класів, що реалізують всю бізнес-логіку web-додатку. Ці класи відповідають за обробку даних (сутностей), розміщення їх в БД, читання з БД, а також за взаємодію між самими об'єктами, складовими таких даних.
- Подання (View) - набір класів і шаблонів, що відповідають за інтерфейс взаємодії з користувачами (User Interface, UI). Зазвичай вони формують HTML-сторінки, що показують користувачам дані з моделі. На основі даних моделі користувачам надається можливість їх переглядати і редагувати.
- Контролер (Controller) - це сполучна ланка між першими двома компонентами. Класи даного компонента отримують дані, що містяться в запиті до сервера (наприклад, значення, отримані з відправленої форми) і передають їх в Модель для обробки і збереження. Після цього Контролер вибирає, яким способом показати їх клієнту за допомогою використання логіки, переданої із моделі, і передає йому дані для формування HTML-сторінки.

Такий поділ web-додатку на частини спрощує структуру програми за рахунок більш строго поділу його рівнів. Логіка для користувача інтерфейсу розміщується в поданні, логіка введення-виведення в контролері, а бізнес-логіка - в моделі. Досягається повне відділення логіки роботи програми від представлення даних. Розробник отримує повний контроль над сформованим HTML - документом. Полегшується завдання виконання тестування програми.

Для реалізації програмних модулів було обрано Angular 6, що базується на партерні MVVM.

2.2 Основна інформація про «Коди швидкої відповіді»

Що таке QR-коди і як вони працюють?



Ви, напевно, бачили QR-коди. Вони є скрізь, від веб-сайтів до реклами. Вони схожі на штрих-коди, але замість вертикальних смуг складаються з квадратних візерунків. Оскільки все більша кількість підприємств починає ними користуватися, вони стають все більш впізнаваними, і багато смартфонів тепер мають вбудовані QR-зчитувачі.



Але що саме це, так звані QR-коди, і як вони працюють? Ось усе, що вам потрібно знати

Визначення: Що означає QR?

QR означає «Швидкий відгук».

Хоча вони можуть виглядати простими QR-кодами, вони насправді здатні зберігати багато даних. Але незалежно від того, скільки вони містять, QR-код

повинен надавати користувачеві миттєвий доступ до інформації - отже, чому він називається кодом швидкого реагування.

Коротка історія QR-коду

Перша система QR-коду була винайдена в 1994 році японською компанією Denso Wave, дочірньою компанією Toyota. Їм потрібен був спосіб точного відстеження транспортних засобів та деталей під час виготовлення. Щоб досягти цього, вони прагнули розробити тип штрих-коду, який міг би кодувати кандзі, кана та буквено-цифрові символи.

Вихід за межі штрих-коду

Звичайні штрих-коди можна читати лише в одному напрямку - зверху вниз. Це означає, що вони можуть зберігати лише невелику кількість інформації, як правило, у буквено-цифровому форматі.

Але QR-код читається у двох напрямках (отже, чому його іноді також називають «двовимірним штрих-кодом») - зверху вниз та праворуч ліворуч. Це дозволяє йому зберігати більший обсяг даних.

Команда розробників QR-коду хотіла полегшити сканування коду, щоб оперативні працівники не витрачали час, отримуючи його під правильним кутом, і вони хотіли, щоб він мав характерний дизайн, щоб його було легко ідентифікувати. Це змусило їх вибрати знакову квадратну форму, яка використовується і сьогодні.

На розробку першого QR-коду пішло більше року. Він міг вміщувати до 7000 цифр плюс символи кандзі, і його можна було прочитати в десять разів швидше, ніж звичайний штрих-код.

QR-код стає відкритим

Після випуску QR-код був швидко прийнятий автопромислом. На той час це було особливо важливо, оскільки багато споживачів в Японії вимагали більшої прозорості від великих корпорацій. Вони хотіли точно знати, звідки походить їхня продукція - і не лише машини, а й їжа, фармацевтика та інші продукти. Тому QR-код був поступово інтегрований в обробну промисловість.

Denso Wave виявилися досить люб'язними, щоб зробити свій QR-код загальнодоступним, заявивши, що вони не будуть використовувати свої патентні права. Це означало, що кожен може створювати та використовувати свої власні QR-коди навіть сьогодні. Однак первинне впровадження ідеї було повільним. Незважаючи на те, що QR-коди самі по собі були легкими у виробництві, QR-читачі, як правило, були недоступні для широкого загалу. Це було до ...

2002: перші мобільні QR-зчитувачі

У 2002 році в Японії були продані перші мобільні телефони, що містять вбудовані QR-зчитувачі. Це призвело до збільшення кількості компаній, що використовують QR-коди, і почали з'являтися перші QR-коди, готові до споживання.

Тим часом технологія продовжувала розвиватися. У 2004 році був створений мікро QR-код, а в 2008 році був випущений код iQR, який мав прямокутні модулі.

Також приблизно в цей час на ринок вийшов перший iPhone, який передав QR-зчитувачі в руки більшої кількості споживачів у всьому світі. Розуміючи потенціал QR-кодів для продажу та маркетингу, бренди та організації почали створювати власні.

Золотий вік QR-кодів

У 2012 році, через вісімнадцять років після свого створення, QR-код був нагороджений нагородою Good Design Award за промисловий дизайн.

На цей час QR-код поширився по всьому світу - його можна було побачити скрізь, від рекламних щитів до журналів до листівок, і в деяких досить незвичних місцях.

Навіть сьогодні, у 2020 році, Denso Wave продовжують вдосконалювати свій оригінальний дизайн. Їхні сучасні QR-коди мають інтегровані рішення,

такі як простежуваність, захист торгової марки та заходи проти підробки. Існує також багато нових способів використання QR-коду - від переказу платежів до визначення позицій об'єктів у доповненій реальності.

Як працюють QR-коди?

Шаблони в QR-кодах представляють двійкові коди, які можна інтерпретувати для виявлення даних, що зберігаються кодом.

Зчитувач QR може ідентифікувати стандартний QR-код на основі трьох великих квадратів на зовнішній стороні QR-коду. Опізнавши ці три фігури, він знає, що все, що міститься всередині квадрата, є QR-кодом.



Потім QR-пристрій аналізує QR-код, розбиваючи все на сітку. Він розглядає окремі квадрати сітки і присвоює кожному значення залежно від того, чорне воно чи біле. Потім він об'єднує квадрати сітки, щоб створити більші візерунки.

Які частини QR-коду?

Стандартний QR-код можна ідентифікувати на основі шести компонентів:

Тиха зона

Це порожня біла рамка навколо зовнішньої частини QR-коду. Без цієї межі зчитувач QR не зможе визначити, що є і що не міститься в QR-коді через перешкоди зовнішніх елементів.

Візерунок шукача

QR-коди зазвичай містять три чорні квадрати в нижньому лівому, верхньому лівому та верхньому правому кутах. Ці квадрати говорять QR-читачеві, що він дивиться на QR-код і де лежать зовнішні межі коду.

Шаблон вирівнювання

Це ще один, менший квадрат, який міститься десь біля нижнього правого кута, що забезпечує можливість читання QR-коду, якщо він перекошений або під кутом.

Шаблон синхронізації

Це Г-подібна лінія, яка проходить між трьома квадратами у візерунку шукача. Шаблон синхронізації допомагає читачеві ідентифікувати окремі квадрати в цілому коді, а також дозволяє пошкоджений QR-код все ще читати.

Інформація про версію

Це невелике поле інформації, що міститься біля верхньої правої комірки шаблону пошуку. Це визначає, яку версію QR-коду читають (див. “Які чотири версії QR-коду?”).

Осередки даних

Решта QR-коду передає фактичну інформацію, тобто URL-адресу, номер телефону або повідомлення, яке він містить.

Які чотири версії QR-коду?

QR-коди можна використовувати для багатьох різних цілей, але, коли мова заходить про типи, існує чотири загальновизнаних «типи». Використана версія визначає спосіб збереження даних і називається “режимом введення”. Він може бути чисельним, буквено-цифровим, двійковим або кандзі. Тип режиму повідомляється через поле інформації про версію в QR-коді.

Числовий режим

Це для десяткових цифр від 0 до 9. Це найефективніший режим зберігання із до 7089 символів.

Буквено-цифровий режим

Це для десяткових цифр від 0 до 9, плюс великі літери від A до Z та символів \$, %, *, +, -, ., /, Та: а також пробілу. Це дозволяє зберігати до 4296 символів.

Режим байтів

Це для символів із набору символів ISO-8859-1. Це дозволяє зберігати 2953 символи.

Режим кандзі

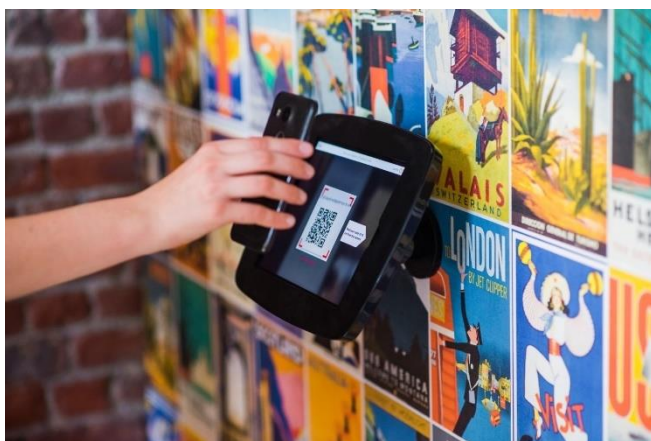
Це для двобайтових символів із набору символів Shift JIS і використовується для кодування символів японською мовою. Це оригінальний режим, вперше розроблений Denso Wave. Однак з тих пір він став найменш ефективним - лише 1817 символів доступні для зберігання.

Існує також другий режим кандзі, який називається режим розширеної інтерпретації каналів (ECI), який визначає набір символів кандзі UTF-8. Однак деякі новіші зчитувачі QR-коду не зможуть прочитати цей набір символів.

Є два додаткові режими, які є модифікаціями інших типів:

- Режим структурованого додавання
- Це кодує дані в кількох QR-кодах, дозволяючи одночасно зчитувати до 16 QR-кодів.
- Режим FNC1

Це дозволяє QR-коду функціонувати як штрих-код GS1.



Чи може QR-код використовувати кілька режимів?

Це можливо, якщо кожен QR-код містить правильне поле інформації про версію.

Які переваги QR-кодів?

QR-код був розроблений для вдосконалення дизайну штрих-коду.

Він може містити більше інформації, ніж штрих-код

Штрих-код може містити лише один рядок інформації у вигляді числового коду. Це робить його корисним у таких середовищах, як роздрібна торгівля, оскільки він може швидко та точно ідентифікувати товар, до якого прикріплений унікальний номер. Однак він не може зробити набагато більше, ніж це.

Тим часом QR-код може містити набагато більше інформації різного типу (наприклад, слова та символи). Він робить це все приблизно в тому ж просторі, що і штрих-код, і його так само легко генерувати та друкувати.

Менше помилок

Ви коли-небудь брали товар на касу і сканували штрих-код, лише щоб машина придумала абсолютно неправильний товар? Це рідкісна обставина, але це трапляється, як правило, оскільки штрих-код не захищений від дублювання. Однак вища здатність зберігати інформацію QR-коду дає можливість вбудувати сейфи.

Легко читається

На відміну від штрих-кодів, для читання та інтерпретації яких потрібна спеціальна технологія, QR-коди можуть читати та розуміти смартфони та цифрові камери. Це робить їх набагато кориснішими для таких речей, як маркетингові кампанії, спрямовані на споживача.

Простота друку

Як і штрих-код, QR-код вимагає лише чорно-білого друку (хоча можна використовувати будь-який колір або навіть кілька кольорів).

Більш безпечний

Можна зашифрувати інформацію в QR-кодах, пропонуючи додатковий рівень захисту.

Які існують різні типи QR-коду?

Можна створювати QR-коди у різних формах та стилях, але найчастіше зустрічаються п'ять типів. Всі вони виконують однакову роботу - просто виглядають дещо інакше.

QR-код

Це оригінальна версія QR-коду, створеного Denso Wave у 1990-х. Це легко визначити за трьома шаблонами пошуку в нижньому лівому, верхньому лівому та верхньому правому кутах.

Ацтекський код



MaxiCode



Цей тип QR-коду використовує поштова служба США. Він схожий на ацтекський код, оскільки він розміщує шаблон посередника посередині, але замість квадратів він використовує шаблон соти.

PDF417



Винайдений у 1991 році Інджиуном Вангом із Symbol Technologies, дивна назва PDF417 передре QR-коду на три роки. Це схоже на поєднання QR-коду та штрих-коду, і його легко впізнати за прямокутною формою.

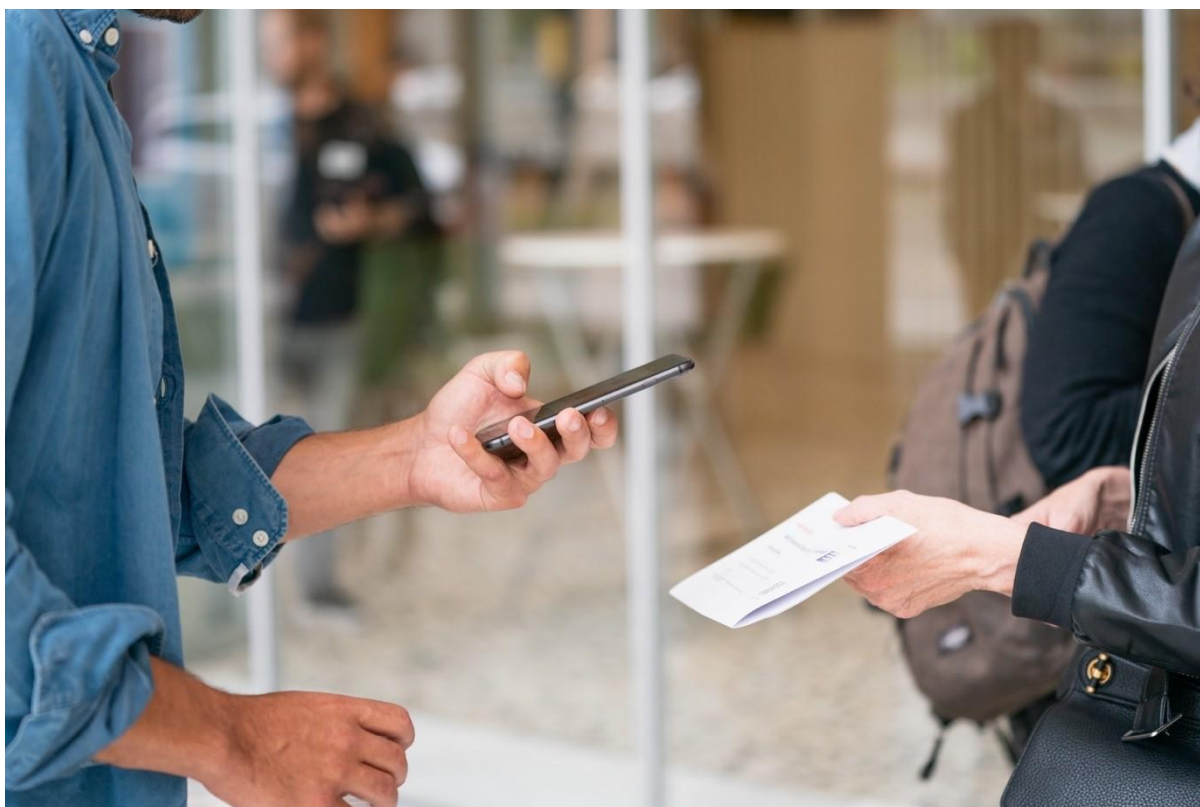
Семако



Розроблений однойменною програмною компанією, Semacode - це матриця даних, яка дуже схожа на звичайний QR-код, але не має впізнаваних шаблонів пошуку.

Як відсканувати QR-код на пристрої iPhone, iPad або Android?

Більшість смартфонів мають вбудовані QR-сканери. Деякі з останніх смартфонів, що вийшли на ринок, такі як найновіший телефон Google Pixel та iPhone, мають вбудовані в камери QR-сканери.



Навіть на деяких планшетах, таких як Apple iPad, у камери вбудовані QR-зчитувачі.

Деякі старі пристрої можуть вимагати спеціального додатка для зчитування QR-кодів, однак таких додатків ви не знайдете в Apple App Store та Google Play.

Сканування QR-коду за допомогою вашого пристрою не може бути простішим:

- Відкрийте програму зчитування QR або камеру на своєму смартфоні
- Наведіть його на QR-код - ви зможете навести камеру під будь-яким кутом і все одно отримаєте необхідну інформацію
- Дані миттєво відобразяться на екрані - наприклад, якщо QR-код містить контактні дані, їх слід негайно завантажити на телефон

Яку інформацію можуть містити QR-коди?

Як і штрих-код, QR-код зберігає дані. Ці дані можуть містити URL-адреси веб-сайтів, номери телефонів або до 4000 символів тексту.

QR-коди також можна використовувати для:

Посилайте безпосередньо, щоб завантажити програму з Apple App Store або Google Play

Аутентифікуйте онлайнві облікові записи та перевірте дані для входу

Отримуйте доступ до Wi-Fi, зберігаючи деталі шифрування, такі як SSID, пароль та тип шифрування

Надсилайте та отримуйте інформацію про оплату

Набагато більше - компанія у Великобританії під назвою QR Memories навіть створює QR-коди для використання на надгробках, дозволяючи людям надсилати повідомлення з-за могили

Де знаходяться QR-коди?

QR-коди спочатку були винайдені, щоб допомогти відстежувати деталі у виробництві автомобілів, і вони все ще використовуються у всій обробній галузі.

Ви також знайдете QR-коди, що використовуються іншими компаніями, яким потрібно пильно стежити за продуктами та матеріалами, такими як будівельна, машинобудівна та роздрібна торгівля.

Вони також використовуються поштовими службами по всьому світу. Оскільки вони містять велику кількість інформації, наприклад, поштові адреси, вони все частіше покладаються на відстеження посилок.

QR-коди тепер також використовуються в коледжах для сприяння взаємодії зі студентами. Вони з'являлися скрізь від класу до бібліотеки, де їх використовують, щоб допомогти студентам знайти книги, які вони шукають.

Приклади використання QR-коду в маркетингових та рекламних кампаніях

Однак місця, де ви, швидше за все, бачите QR-коди у своєму повсякденному житті, відсутні в продажах та маркетингових кампаніях.

Багато рекламодавців люблять використовувати QR-коди у своїх кампаніях, оскільки це забезпечує швидший та інтуїтивніший спосіб спрямовувати людей на веб-сайти, ніж шляхом введення URL-адрес вручну.

Їх навіть можна використовувати для прямих посилань на сторінки продуктів в Інтернеті. Наприклад, якщо ви шукали саме плаття, на якому модель була одягнене в плакат, QR-код може перейти безпосередньо на веб-сторінку, де ви можете його придбати.

Ви також можете знайти QR-коди на упаковці для деяких своїх улюблених продуктів. Ці QR-коди можуть розкрити інформацію про продукт, таку як харчова інформація, або спеціальні пропозиції, які ви зможете використати для наступного разу.



Ви можете розмістити QR-код де завгодно. Тож не дивуйтеся, якщо ви побачите, що вони з'являються скрізь від візитних карток, до покажчиків та навіть квитанцій.

Як створити QR-код?

Створення QR-коду набагато простіше, ніж ви можете подумати.

Десятки генераторів QR-кодів можна завантажити для вашого смартфона через Apple App Store та Google Play.

Ви також можете створити їх в Інтернеті, використовуючи ряд веб-сайтів, таких як:

- qr-code-generator.com
- the-qrcode-generator.com
- qrcode-monkey.com

Переваги використання генератора QR-коду полягають у тому, що ви можете легко створити QR-код на основі введеної інформації (будь то URL-адреса, контактна картка, подія календаря або текст), налаштувати дизайн різними кольорами та миттєво завантажити та поділитися файлом зображення.

Отримавши дизайн QR-коду, ви зможете відтворити його за допомогою різних носіїв. Поки форму QR-коду може читати і розуміти смартфон, ви можете зробити QR-код майже з усього.

Творчі способи створення QR-кодів

У Чженчжоу, Китай, понад 2500 студентів з Міжнародного коледжу Сіас зібрались у 2017 році, щоб сформувати найбільший у світі QR-код людини. Студенти допомагають парасолькам створювати візерунок розміром 51 метр в поперечнику. При скануванні студенти отримали спеціальну пропозицію від компанії Rideshare Didi-Express.

Також у Китаї, в селі Сіліньшуй, понад 120000 дерев ялівцю було висаджено на 12 гектарах у формі QR-коду, який можна було переглянути зверху. Після сканування він спрямовував користувачів на перегляд офіційної туристичної сторінки Xilinshui.

Багато компаній придумали інші креативні місця для створення QR-кодів. Ось декілька найбільш образних способів, як ми бачили, як люди використовують QR-коди:

- Як штамп
- На футболках
- На різдвяні подарункові теги
- QR-код - поширені запитання

Чи повинні QR-коди бути чорно-білими?

Ні, насправді вони можуть бути будь-якого кольору, якщо контраст між світлими і темними областями чіткий. Насправді QR-коди можуть бути навіть різнокольоровими.

Чи багато людей використовують QR-коди?

Дані свідчать про те, що використання QR-коду збільшується. За даними Сканової та Statista, за оцінками, 11 мільйонів домогосподарств сканують QR-код у 2020 році - це значне збільшення порівняно з 9,76 мільйонами сканувань у 2018 році.

Це пов'язано з трьома основними причинами: проникнення смартфонів зросло з 10% у 2014 році до 36% у 2018 році, тоді як високошвидкісний

мобільний Інтернет зріс із 48% у 2014 році до 61% у 2018 році. Також багато нових смартфонів тепер мають QR-сканери вбудований, що дає більше людей можливість сканувати QR-коди, коли вони позаду.

Чи безпечні QR-коди?

Оскільки QR-коди містять багато даних, цілком можливо, що їх можна використовувати для приховування фішингової інформації.

Розумні хакери також можуть використовувати QR-коди, щоб спрямовувати користувачів на замасковане завантаження шкідливого програмного забезпечення або змусити мобільний телефон користувача надсилати преміум-текстові повідомлення.

А у випадку з QR-кодами, які вказують на URL-адреси, існує можливість зламати домен, до якого вони спрямовують користувачів. В одному з відомих прикладів німецька компанія з виробництва приправ, яка залишатиметься неназваною, випадково спрямовувала користувачів, які сканували свій QR-код на веб-сайт для дорослих.

Як захиститися від зловмисних QR-кодів

Невідомо, де і коли ви можете натрапити на заражений QR-код. Ось чому важливо вибрати QR-сканер, якому, як ви знаєте, можна довіряти.

Kaspersky QR Scanner миттєво перевіряє безпеку відсканованого посилання, перш ніж подавати будь-яку інформацію вам, користувачеві.

Таким чином ви знаєте, чи є за QR-кодом небезпеки, наприклад:

Фішинг-шахрайство

- Примусове завантаження програми або шахрайство з текстовими повідомленнями преміум-класу
- Небезпечні посилання
- QR-сканер Kaspersky все ще надає все необхідне від QR-сканера, наприклад, додавання контактів до телефону. Але він також створює журнал попередніх сканувань, щоб ви могли простежити назад, щоб побачити, коли і де вас могли скомпрометувати.

2.2 Огляд архітектури Angular 8



Рис. 2.4 Структура Angular

Angular - це платформа і основа для побудови клієнтських додатків у HTML і TypeScript. Angular написаний на мові програмування TypeScript . Платформа

Angular реалізує основні та додаткові функції як набір бібліотек TypeScript, які імпортуються у власну програму.

Основними будівельними блоками програми Angular є NgModules, які забезпечують контекст компіляції для компонентів. NgModules збирають відповідний код у функціональні набори ; додаток визначається набором NgModules. Додаток завжди має принаймні кореневий модуль, який дозволяє завантажувати початкову програму, і зазвичай має багато інших модулів.

Компоненти визначають види, які є набором елементів екрану, які Angular може вибрати і змінити відповідно до логіки програми та даних.

Компоненти використовують послуги, які забезпечують певну функціональність, безпосередньо не пов'язану з переглядами. Постачальники послуг можуть бути введені в компоненти як залежності, що робить код модульним, багаторазовим і ефективним.

Як компоненти, так і служби - це просто класи, де декоратори позначають їх тип і надають метадані, які повідомляють Angular, як їх використовувати.

Метадані для класу компонентів асоціюють його з шаблоном, який визначає подання. Шаблон поєднує звичайний HTML з директивами Angular та розміткою, що дозволяє Angular змінити HTML перед його відображенням.

Метадані для класу послуг надають інформацію Angular, яка повинна зробити її доступною для компонентів через ін'єкцію залежностей (DI).

Компоненти програми зазвичай визначають багато переглядів, організованих ієрархічно. Angular надає послугу маршрутизатора, щоб допомогти вам визначити шляхи навігації серед переглядів. Маршрутизатор забезпечує складні навігаційні можливості браузера.

Також в прикладі вище встановлюється властивість `selector`, яке визначає селектор CSS. В елемент з цим селектором Angular буде додавати уявлення компонента. Наприклад, в прикладі вище селектор має значення `my-app`. Відповідно якщо html-сторінка містить елемент `<my-app> </ my-app>`, наприклад:

2.3 Модулі

Модулі Angular `NgModules` доповнюють модулі JavaScript (ES2015). `NgModule` оголошує контекст компіляції для набору компонентів, призначених домену, робочому процесу або тісно пов'язаному набору можливостей. `NgModule` може асоціювати свої компоненти з відповідним кодом, таким як послуги, для формування функціональних одиниць.

Кожен додаток Angular має кореневий модуль, який умовно називається `AppModule`, який забезпечує механізм завантаження та запуску програму.

Подібно модулям JavaScript, `NgModules` можуть імпортувати функціональні можливості інших `NgModules` і дозволяють експортувати та використовувати їхні власні функціональні можливості іншими `NgModules`. Наприклад, щоб використовувати службу маршрутизатора у вашому додатку, ви імпортуєте маршрутизатор `NgModule`.

Упорядкування коду в окремі функціональні модулі допомагає в управлінні розробкою складних додатків, а також у розробці для повторного використання. Крім того, ця методика дозволяє скористатися перевагою лінивого завантаження (lazy-loading), тобто завантаження модулів на вимогу, щоб мінімізувати кількість коду, який необхідно завантажувати під час запуску, що є дуже важливим для швидкодії програми. Тобто під час запуску програми, завантажується лише основна частина, всі ж інші модулі завантажуються, коли в них є потреба.

2.4 Компоненти

Щоб клас міг використовуватися в інших модулях, він визначається з ключовим словом `export`. У самому ж класі визначена лише одна змінна, яка в якості значення зберігає деяку рядок.

Для створення компонента необхідно імпортувати функцію декоратора `@Component` з бібліотеки `@angular / core`. Декоратор `@Component` дозволяє ідентифікувати клас як компонент.

Якби ми не застосували декоратор `@Component` до класу `AppComponent`, то клас `AppComponent` компонентом б не зважав.

Декоратор `@Component` як параметр приймає об'єкт з конфігурацією, яка вказує фреймворку, як працювати з компонентом і його поданням.

За допомогою властивості `template`. Шаблон являє шматок розмітки HTML з вкрапленнями коду Angular. Фактично шаблон це і є уявлення, яке побачить користувач при роботі з додатком.

Кожен компонент повинен мати один шаблон. Однак необов'язково визначати шаблон безпосередньо за допомогою властивості `template`. Можна винести шаблон у зовнішній файл з розміткою `html`, а для його підключення використовувати властивість `templateUrl`.

Шаблон може бути однорядковим або багаторядковим. Якщо шаблон багатостроковий, то він полягає в косі лапки (```), які варто відрізнити від стандартних ординарних лапок (`'`).

2.5 Шаблони, директиви та прив'язка даних

Директиви визначають набір інструкцій, які застосовуються при рендеринге html-коду. Директива представляє клас з директивними метаданими. У TypeScript для прикріплення метаданих до класу застосовується декоратор `@Directive`.

У Angular є три типи директив:

- Компоненти: компонент по суті також є директивою, а декоратор `@Component` розширює можливості декоратора `@Directive` за допомогою додавання функціоналу по роботі з шаблонами.
- Атрибутивні: вони змінюють поведінку вже існуючого елемента, до якого вони застосовуються. Наприклад, `ngModel`, `ngStyle`, `ngClass`
- Структурні: вони змінюють структуру DOM за допомогою додавання, зміни або видалення елементів html. Наприклад, це директиви `ngFor` і `ngIf`

У секції `styles` у компонента визначено два класи, які встановлюють різні стильові властивості шрифту: `verdanaFont` і `segoePrintFont`. У шаблоні для прив'язки класу до елемента застосовується директива `[ngClass] = "{verdanaFont: true}"`. Ця директива приймає js-об'єкт, в якому ключі - це назви класів. Цим назвами присвоюються булеві значення `true` (якщо клас застосовується) і `false` (якщо клас не застосовується). Тобто в даному випадку клас `verdanaFont` буде застосовуватися до всього блоку `div`. Однак в блоці `div` є параграф, і ми, припустимо, хочемо, щоб до цього параграфу застосовувався інший клас. А за замовчуванням вкладений параграф успадкує стилі від батьківського блоку `div` і також застосовує клас `segoePrintFont`, в якому можна перевизначити успадковані стилі.

```
import { Component } from '@angular/core';

@Component({
  selector: 'my-app',
  template: `<div [ngClass]="{verdanaFont:isVerdana}">
```

```

        <h1>Hello Angular</h1>
        <p [ngClass]="{segoePrintFont:isSegoe}">
            Angular представляє модульну архітектуру приложения
        </p>
    </div>`,
    styles: [
        `.verdanaFont{font-size:13px; font-family:Verdana;}`
        `.segoePrintFont{font-size:14px; font-family:"Segoe Print";}`
    ]
  })
  export class AppComponent {
    isVerdana = true;
    isSegoe = true;
  }

```

2.6 Послуги та ін'єкції залежності

Для даних або логіки, які не пов'язані з певним переглядом, і які потрібно спільно використовувати для компонентів, ви створюєте клас послуг. Визначення класу служб безпосередньо передує декоратору `@Injectable ()`. Декоратор надає метадані, які дозволяють іншим постачальникам вводитись як залежність у ваш клас. Ін'єкція залежностей (DI) дозволяє зберігати класи класів компонентів та ефективні. Вони не отримують дані з сервера, перевіряють вхід користувача або безпосередньо входять до консолі; вони делегують такі завдання службам.

2.7 Маршрутизація

Кутовий маршрутизатор `NgModule` надає послугу, яка дозволяє визначити шлях навігації між різними станами програм і переглядати ієрархії у вашому додатку. Він моделюється за звичними умовами навігації браузером: введіть URL-адресу в адресному рядку, а браузер перейде до відповідної сторінки. Натисніть посилання на сторінці, і браузер перейде на нову сторінку. Натисніть кнопку назад і вперед браузера, і браузер переходить назад і вперед по історії сторінок, які ви бачили. Маршрутизатор відображає URL-шляхи до переглядів замість сторінок. Коли користувач виконує певну дію, наприклад, натиснувши на посилання, яка б завантажувала нову сторінку у веб-переглядачі, маршрутизатор перехоплює поведінку веб-переглядача і показує ієрархії

перегляду. Якщо маршрутизатор визначає, що поточний стан програми вимагає особливих функціональних можливостей, а модуль, який визначає його, не був завантажений, маршрутизатор може зтягувати модуль за вимогою. Маршрутизатор інтерпретує URL-адресу посилання відповідно до правил навігації перегляду вашої програми та стану даних. Ви можете перейти до нових переглядів, коли користувач натискає кнопку або вибирає з вікна випуску, або у відповідь на інший стимул з будь-якого джерела. Маршрутизатор реєструє активність в історії браузера, тому також працюють кнопки назад і вперед. Щоб визначити правила навігації, ви пов'язуєте шляхи навігації з компонентами. Шлях використовує синтаксис, подібний до URL, який інтегрує дані програми, так само, як синтаксис шаблону інтегрує ваші перегляди з даними вашої програми. Потім можна застосувати логіку програми, щоб вибрати, які види відображати або ховати, у відповідь на вхід користувача та власні правила доступу.

Основи основних будівельних блоків програми Angular. Наведена нижче схема показує, як ці основні частини пов'язані між собою.

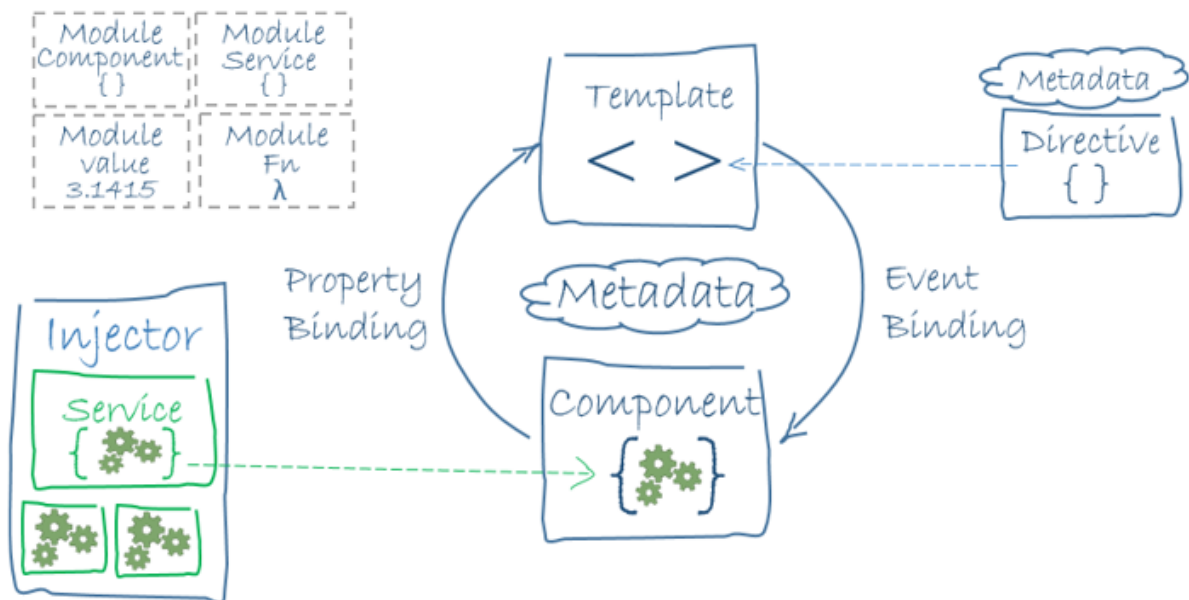


Рис. 2.5 Логічне представлення Angular

Разом компонент і шаблон визначають вид Angular.

Декоратор класу компонентів додає метадані, включаючи показник на асоційований шаблон.

Директиви та розмітка прив'язки в шаблоні компонента змінюють види на основі даних програми та логіки.

Інжектор залежностей надає послуги компоненту, наприклад службу маршрутизатора, яка дозволяє визначати навігацію між переглядами.

2.8 Введення в модулі

Angular програми є модульними, а Angular має власну систему модульності під назвою NgModules. NgModules є контейнерами для когезивного блоку коду, присвяченого домену додатку, робочому процесу або тісно пов'язаному набору можливостей. Вони можуть містити компоненти, постачальники послуг та інші кодові файли, область дії яких визначається вмістом NgModule. Вони можуть імпортувати функціональні можливості, що експортуються з інших NgModules, і експортувати вибрані функціональні можливості для використання іншими NgModules.

Кожен додаток Angular має принаймні один клас NgModule, кореневий модуль, який умовно називається AppModule і знаходиться у файлі з ім'ям app.module.ts. Ви запускаєте свою програму, завантаживши кореневу версію NgModule.

Хоча невелика програма може мати лише один NgModule, більшість програм мають багато інших функціональних модулів. Корінний NgModule для програми називається так, тому що він може включати дочірні NgModules в ієрархії будь-якої глибини.

2.9 Метадані NgModule

NgModule визначається класом, прикрашеним @NgModule (). Декоратор @NgModule () - це функція, яка приймає один об'єкт метаданих, властивості якого описують модуль. Найважливішими властивостями є наступні.

декларації: компоненти, директиви та труби, що належать до цього NgModule.

export: Підмножина декларацій, які повинні бути видимими і придатними для використання в шаблонах компонентів інших NgModules.

import: Інші модулі, експортовані класи необхідні шаблонами компонентів, оголошеними в цьому NgModule.

Провайдери: творці послуг, які як і цей NgModule сприяють глобальному збору послуг; вони стають доступними у всіх частинах програми. (Ви також можете вказати постачальників на рівні компонентів, який часто є кращим.)

Bootstrap: Головний вигляд програми, званий кореневим компонентом, який розміщує всі інші види програм. Тільки властивість `root` NgModule повинна встановити властивість `bootstrap`.

Ось просте визначення кореневого NgModule.

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
@NgModule({
  imports: [ BrowserModule ],
  providers: [ Logger ],
  declarations: [ AppComponent ],
  exports: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

2.10 NgМодулі та компоненти

NgModules надають контекст компіляції для своїх компонентів. Кореневий NgModule завжди має кореневий компонент, який створюється під час завантаження, але будь-який NgModule може включати будь-яку кількість додаткових компонентів, які можна завантажити через маршрутизатор або створити за допомогою шаблону. Компоненти, що належать до NgModule, мають спільний контекст компіляції.

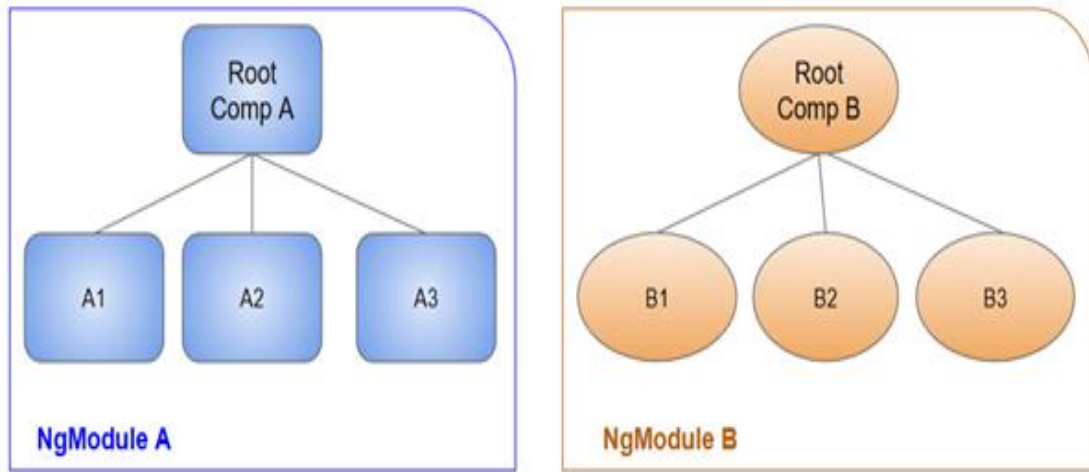


Рис.2.6 Схема зв'язку NgModules

Компонент і його шаблон разом визначають вигляд. Компонент може містити ієрархію виду, яка дозволяє визначати довільно складні області екрану, які можна створювати, змінювати і знищувати як одиницю. Ієрархія перегляду може змішувати види, визначені в компонентах, які належать до різних NgModules. Це часто буває, особливо для бібліотек інтерфейсу.

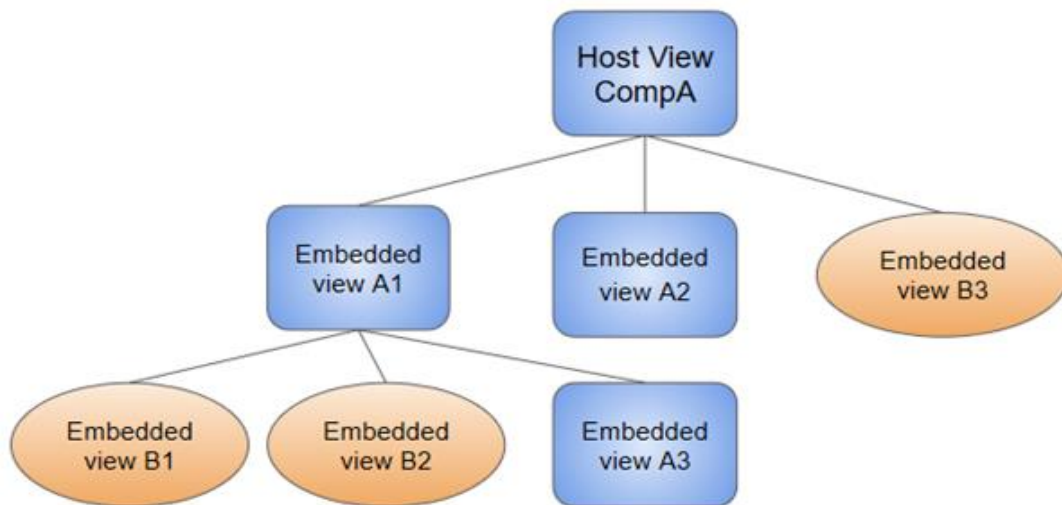


Рис. 2.7 Схема зв'язку бібліотек патерну.

Коли ви створюєте компонент, він безпосередньо пов'язаний з одним переглядом, який називається вікном хосту. Вигляд вузла може бути коренем ієрархії перегляду, яка може містити вбудовані подання, які, у свою чергу, відображають хост інших компонентів. Ці компоненти можуть бути в одному NgModule, або можуть бути імпортовані з інших NgModules. Перегляди в дереві можуть бути вкладені на будь-яку глибину.

2.11 Модулі NgModules і JavaScript

Система NgModule відрізняється і не пов'язана з модульною системою JavaScript (ES2015) для управління колекціями об'єктів JavaScript. Це додаткові модульні системи, які можна використовувати разом для запису додатків.

У JavaScript кожен файл є модулем, і всі об'єкти, визначені у файлі, належать до цього модуля. Модуль оголошує деякі об'єкти публічними, позначивши їх ключовим словом експорту. Інші модулі JavaScript використовують оператори імпорту для доступу до загальнодоступних об'єктів з інших модулів.

```
import { NgModule } from '@angular/core';  
import { AppComponent } from './app.component';  
export class AppModule { }
```

2.12 Angular libraries

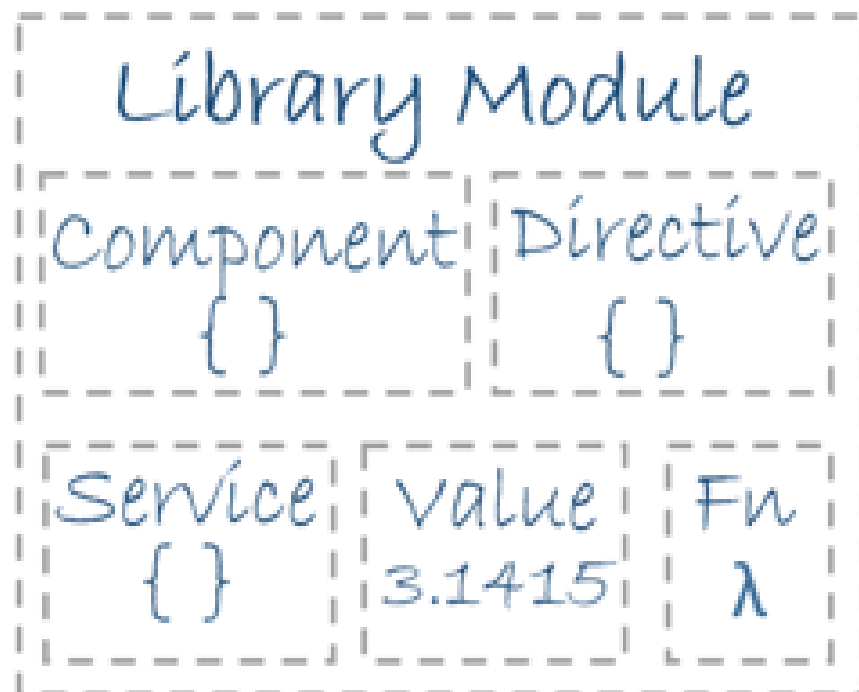


Рис. 2.8 Представлення Angular libraries

Angular навантаження як збірка модулів JavaScript. Ви можете вважати їх бібліотечними модулями. Кожне ім'я бібліотеки Angular починається з @angular префікса. Встановіть їх за допомогою менеджера пакетів npm і імпортуйте їх з операторами імпорту JavaScript.

Наприклад, імпортувати декоратор компонентів Angular з бібліотеки @angular / core подібно.

```
import { Component } from '@angular/core';
```

Ви також імпортуєте NgModules з кутових бібліотек за допомогою операторів JavaScript import. Наприклад, наступний код імпортує BrowserModule NgModule з бібліотеки браузера платформи.

```
import { BrowserModule } from '@angular/platform-browser';
```

У прикладі простого кореневого модуля, описаного вище, модуль програми потребує матеріалу в межах BrowserModule. Щоб отримати доступ до цього матеріалу, додайте його до імпорту метаданих @NgModule, наприклад.

```
imports: [ BrowserModule ],
```

Таким чином, використовуються модульні системи Angular і JavaScript разом. Хоча можна легко переплутати дві системи, які поділяють загальний словник "імпорту" та "експорту".

2.13 Dotnet core

З'явившись в 2002 році, платформа .NET пройшла довгий шлях і змогла завоювати сильні позиції як надійне рішення корпоративного рівня. Довго час .NET була мультиплатформенною середовищем, що дозволяє створювати серверні і десктопні програми, програми для мобільних і вбудованих пристроїв, додатки для розгортання в хмарної інфраструктурі. Але при цьому не була кроссплатформенною і підтримувала виключно операційні системи сімейства Windows.

Швидше за все, якщо ви чули про .NET Core, то чули про нього те, що це новий .NET, який працює під Linux і Mac OS X. Саме можливість запуску платформи на ОС, відмінних від Windows, і викликала свого часу безліч

суперечок і обговорень. Хоча, насправді, задовго до появи .NET Core вже існували Кроссплатформені реалізація .NET Framework. Два найвідоміших проекту - відомий проект Mono, який не раз був відзначений навіть самої Microsoft, і DotGNU, свого часу підтримуваний Free Software Foundation. На сьогоднішній день проект DotGNU вже закритий, а ось Mono, навпаки, в останні два роки отримав активний розвиток. Mono являє собою open source реалізацію .NET, що підтримує операційні системи Linux і Mac OS X. Розвивається Mono незалежним спільнотою розробників, які займаються реінжинирингом компонентів .NET і створюють їх кроссплатформенну реалізацію. Зважаючи на це Mono завжди була «наздоганяючою» платформою, в якій можливості оригінального .NET з'являлися через досить тривалий час.

Головна відмінність .NET Core від Mono полягає в тому, що Mono - це саме перенесення «великого» .NET, на платформу * nix. У той час, як .NET Core - це спроектована практично з нуля платформа, спочатку розрахована на роботу з різними ОС. При цьому велика частина коду якою писалася з тим, щоб переносних специфічних залежностей було якомога менше. На наведеному нижче графіку - співвідношення загального коду платформи і коду специфічного для кожного окремо взятого сімейства ОС:

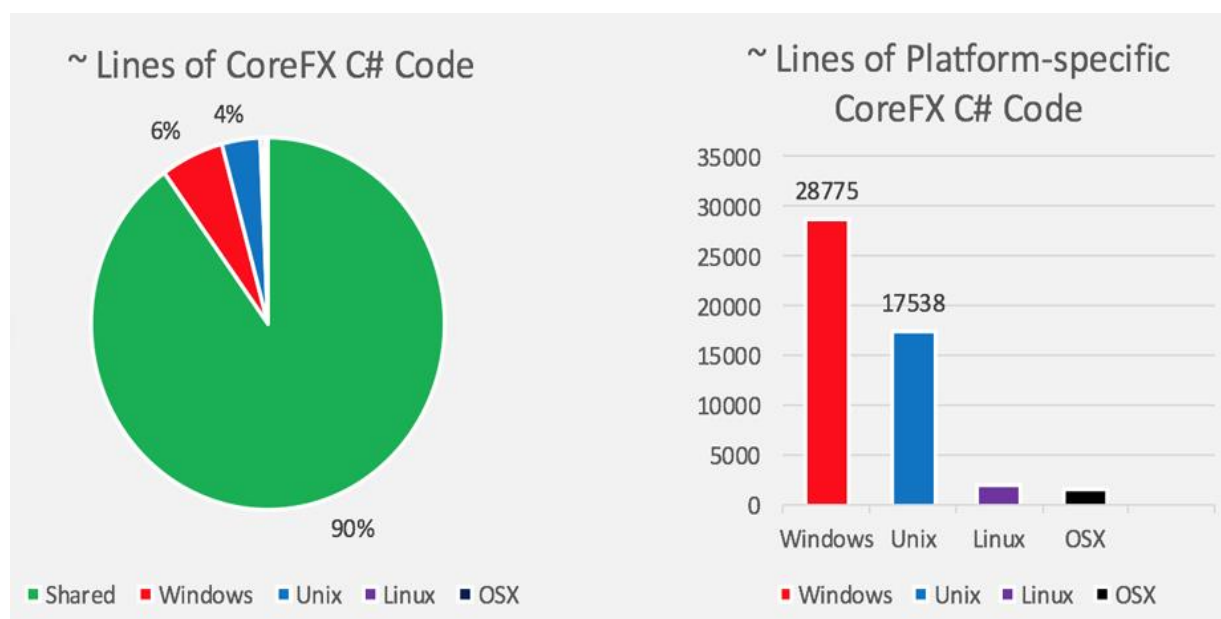


Рис.2.9 Діаграма використання .NET

Open Source. На відміну від класичного .NET Framework, код якого здебільшого є закритим, код .NET Core повністю відкритий і розповсюджується під ліцензіями MIT і Apache2.

Варто відзначити, що після відкриття коду .NET Core команда, що займається розробкою проекту Mono, оголосила про намір об'єднання кодової бази тих компонентів Mono, які реалізовані в .NET Core.

Робота в хмарі. Як і проект, написаний на класичному .NET, проект на базі .NET Core досить легко перенести в хмару. Microsoft Azure вже підтримує розміщення .NET Core проектів як в службах Application Services, так і на віртуальних машинах.

Бібліотеки для роботи з сервісами Microsoft Azure також вже починають перенести на .NET Core. Наприклад, Windows Azure Storage вже доступна для роботи. Відповідно Azure Storage Services можуть використовуватися в core-проектах.

Також, для проектів .NET Core з'являється можливість розміщення на майданчиках тих хмарних провайдерів, які не забезпечили підтримку Windows оточення, але при цьому мають інші привабливими можливостями. Наприклад, віртуальні сервери Digital Ocean за рахунок використання SSD дисків дуже швидкі, але на даний момент дозволяють розгортати тільки ОС сімейства Linux. Розгорнути проект на .NET Core на подібно сервері не складе труднощів.

2.14 Інструменти для роботи з .NET Core

Незважаючи на те, що платформа тільки недавно отримала статус RTM (release to market), вже є ряд зручних і потужних засобів для розробки. Давайте розглянемо докладніше, які інструменти доступні вже зараз для .NET Core розробників.

Project Rider - IDE від компанії JetBrains, тієї самої компанії, яка створила ReSharper, напевно, найпопулярніший плагін для Visual Studio. В основі Rider лежать IntelliJ IDEA (що є основною для цілого сімейства IDE) і

напрацювання, які використовуються в ReSharper. На даний момент Rider підтримує розробку як під Mono, так і під .NET Core. Слід пам'ятати, що на даний момент Rider знаходиться в стадії EAP (Early Access Preview) і по суті не призначений для використання в production, однак вже дає можливість оцінити наявний потенціал. Rider доступний як під Windows, так і під Linux і Mac OS X.

Visual Studio Code - редактор (хоча вже практично IDE), розроблений компанією Microsoft на базі ядра Electron (того самого, на якому заснований Atom). Завдяки великій кількості плагінів VS Code підтримує розробку не тільки під Mono і .NET Core, але також і під іншими мовами, включаючи Go, C / C ++, JavaScript, Typescript, etc. Visual Studio Code працює як під Windows, Linux і Mac OS X.

2.15 Існуючі рішення

Platformus CMS. «Platformus CMS - молода система управління вмістом веб-сайтів (10-я альфа на момент написання статті), побудована на базі не менше молодих ASP.NET Core і ExtCore Framework. Написана CMS на C #. Завдяки можливостям ASP.NET Core, вона однаково добре може працювати на Windows, Linux і Mac. Сама виконується середовище, необхідне для роботи будь-якої програми на .NET Core, може бути як встановлена окремо, так і інтегрована безпосередньо в сам додаток ».

Orchard. Orchard - система управління контентом з відкритим вихідним кодом від компанії Microsoft. Перша версія CMS була анонсований в березні 2010 року і базувалася на ASP.NET MVC. Друга версія активно розробляється вже на основі ASP.NET Core.

2.16 Платформа корпоративного рівня для всіх

Зазвичай, коли говорять про .NET, то в першу чергу мають на увазі великі корпоративні проекти з високими вимогами до надійності, масштабованості і розширюваності. І при цьому з дорогою інфраструктурою. Тому розробка під

.NET завжди була досить недешевим задоволенням. Однак з .NET Core все трохи інакше.

Залишаючи в наявності всі плюси класичної версії, нова версія дозволяє значно знизити інфраструктурні витрати. Хостинг проекту на .NET Core обходиться набагато дешевше хостингу проекту на «великому» .NET. Так в Microsoft Azure найдешевша віртуальна машина з можливістю хостингу ASP.NET веб-додатки буде коштує від \$ 13 в місяць. Розміщення в службі Azure App Services в тарифі Shared буде коштувати від \$ 9 на місяць. При цьому розміщення проекту на віртуальному сервері у Digital Ocean обійдеться всього в \$ 5 в місяць. При цьому кількість ресурсів, що виділяються на додаток буде набагато більше, ніж у Shared-сервісу, а якщо порівнювати з віртуальною машиною від Azure, то конфігурація Digital Ocean буде вигідно відрізнятися наявністю SSD диска.

Звичайно, для великих корпоративних проектів подібні суми не мають ніякого значення. Більш того, корпоративне середовище досить консервативна. Тому, на мій погляд, .NET Core почне широко використовуватися в enterprise проектах не раніше ніж через 2-3 роки, як це було з ASP.NET MVC. У поточному ж своєму вигляді нова платформа буде грати в тому ж сегменті, де зараз знаходяться такі технології, як Node.js і Ruby - це невеликі проекти з обмеженим бюджетом і невисокою архітектурної складністю. Великі рішення як і раніше будуть реалізовуватися на «великому». NET. Таким чином, цільовою аудиторією .NET є стартапи і ринок малого і середнього бізнесу.

Висновки: виходячи з усього вищезазначеного, можна зробити висновок, що .NET Core дозволяє невеликим проектам і стартапам отримати всі переваги платформи корпоративного рівня, при цьому надаючи зручні та засоби розробки, а також недорогу інфраструктуру. В майбутньому ж неминучий прихід платформи і на великий корпоративний ринок. В моєму проекті використано саме .NET Core оскільки він дає змогу легкого симбіозу front-end та back-end частин проекту, що відповідає нашим потребам.

2.17 Azure

Примітки до журналу аудиту

Для файлів журналів у форматі JSON MySQL Enterprise Audit підтримує операції зчитування журналу за допомогою визначеної користувачем функції `audit_log_read ()`. Раніше вказати позицію, з якої починати читання, можна було лише передавши в `audit_log_read ()` аргумент, що містить закладку із зазначенням точної позначки часу та ідентифікатора події певної події. Для більшої гнучкості аргументом тепер може бути специфікатор запуску, який називає будь-яку позначку часу, щоб прочитати її, починаючи з першої події, що відбувається в цій позначці часу або після неї. Див. Читання журналів аудиту.

Примітки C API

Клієнтська бібліотека MySQL тепер включає в себе функцію API `mysql_real_connect_dns_srv ()`, яка схожа на `mysql_real_connect ()`, але використовує запис DNS SRV для визначення хостів-кандидатів для встановлення підключення до сервера MySQL, а не явних аргументів хосту, порту та сокета. .

Додатки, що використовують API C, можуть викликати нову функцію безпосередньо. Крім того, клієнтська програма `mysql` модифікована для використання можливостей DNS SRV; тепер він підтримує параметр `--dns-srv-name`, який має перевагу над `--host` і викликає підключення на основі запису DNS SRV. Див. `Mysql_real_connect_dns_srv ()`.

Встановлення з'єднання в інших контекстах не впливає, включаючи з'єднання, здійснені репліками, FEDERATED механізмом зберігання та клієнтськими програмами, крім `mysql`.

Примітки до компіляції

Visual Studio 16.4 тепер є мінімальною версією для компіляції MySQL. (Помилка # 31655401)

Зараз мінімальна версія бібліотеки Boost для збірки серверів становить 1.73.0. (Помилка # 31309800)

Примітки до конфігурації

Нова опція `WITH_TCMALLOC` CMake вказує, чи потрібно пов'язувати з `-ltcmalloc`. Якщо ввімкнено, вбудовані підпрограми `malloc ()`, `calloc ()`, `realloc ()` та `free ()` вимикаються. За замовчуванням `OFF`. `WITH_TCMALLOC` і `WITH_JEMALLOC` взаємовиключні. (Помилка # 31785166)

Нова опція `COMPRESS_DEBUG_SECTIONS` CMake вказує, чи потрібно стискати розділи налагодження двійкових виконуваних файлів (лише для Linux). Стискання виконуваних розділів налагодження економить простір за рахунок додаткового часу процесора під час процесу збірки. За замовчуванням `OFF`. Якщо цей параметр не встановлений явно, але встановлено змінну середовища `COMPRESS_DEBUG_SECTIONS`, параметр отримує своє значення з цієї змінної. (Помилка # 31498296)

Параметр `WITH_DEFAULT_FEATURE_SET` CMake видалено. (Помилка # 31122507)

На платформах, що реалізують підтримку простору імен мережі (наприклад, Linux), MySQL тепер дозволяє налаштовувати мережевий простір імен для підключень TCP / IP від клієнтських програм до сервера MySQL або плагіна X:

На стороні сервера системні змінні `bind_address`, `admin_address` та `mysqlx_bind_address` мають розширений синтаксис для визначення простору імен мережі, який використовуватиметься для даної IP-адреси або імені хосту, на якому слід прослуховувати вхідні з'єднання.

Для клієнтських з'єднань клієнт `mysql` та клієнт тестового набору `mysqlxtest` підтримують опцію `--network-namespace` для зазначення простору імен мережі.

Для з'єднань реплікації із серверів-реплік до вихідних серверів оператор `CHANGE MASTER TO` підтримує опцію `NETWORK_NAMESPACE` для вказівки простору імен мережі.

Для цілей моніторингу реплікації використовується таблиця продуктивності `replication_connection_configuration`, сховище метаданих підключення до сервера реплікації (див. Репозиторії метаданих реплікації) та `SHOW REPLICA` | Інструкція `SLAVE STATUS` має новий стовпець, що відображає застосовний простір імен мережі для з'єднань.

Для отримання додаткової інформації, включаючи передумови системи хосту, які повинні бути виконані для використання цієї функції, див. Розділ Підтримка простору імен мережі.

Примітки щодо припинення дії та вилучення

Плагін InnoDB memcached застарілий, і підтримка його буде вилучена в майбутній версії MySQL.

Таблиця `INFORMATION_SCHEMA.TABLESPACES` не використовується. Зараз він застарів і буде вилучений у наступній версії MySQL. Інші таблиці `INFORMATION_SCHEMA` можуть містити відповідну інформацію, як описано в Таблиці `INFORMATION_SCHEMA.TABLESPACES`.

Примітки до брелоків

MySQL Enterprise Edition тепер включає плагін `keyring_osi`, який використовує сховище інфраструктури Oracle Cloud як задній кінець для зберігання ключів. Жодна ключова інформація постійно не зберігається в локальному сховищі сервера MySQL. Усі ключі зберігаються у сховищі Oracle Cloud Infrastructure Vault, завдяки чому цей плагін добре підходить для клієнтів Oracle Cloud Infrastructure MySQL для управління ключами MySQL Enterprise Edition. Для отримання додаткової інформації див. Розділ Блокування ключів MySQL.

Примітки оптимізатора

Важлива зміна: Підготовлене твердження тепер готується лише один раз під час виконання `PREPARE`, а не один раз кожного разу, коли воно виконується. Крім того, заява всередині збереженої процедури також готується лише один раз, коли збережена процедура виконується вперше. Ця зміна покращує ефективність таких тверджень, оскільки дозволяє уникнути додаткових витрат на повторну підготовку та відкат структур підготовки, останній є джерелом кількох помилок.

В рамках цієї роботи змінюється спосіб вирішення динамічних параметрів, використаних у підготовлених виписках, з результатами змін у випадках використання підготовлених виписок, перелічених тут:

Параметр, який використовується у підготовленому операторі, визначає тип даних, коли оператор готується, і тип зберігається для кожного наступного виконання оператора, якщо тільки оператор не є типовим (див.

Для підготовленого твердження форми `SELECT expr1, expr2, ... FROM table ORDER BY?`, Передача цілого значення `N` для параметра більше не спричинює упорядкування результатів за `N`-м виразом у списку вибору; результати більше не впорядковуються, як це очікується при константі `ORDER BY`.

Функції вікна `NTILE (NULL)`, `NTH_VALUE (expr, NULL)`, `LEAD (expr, nn)` та `LAG (expr, nn)`, де `nn` є від'ємним числом, тепер заборонені, щоб відповідати стандарту SQL.

Користувацька змінна, яка читається підготовленим випискою, тепер визначає тип, коли виписується виписка; тип зберігається для кожного наступного виконання оператора.

Користувацька змінна, яка читається в операторі в межах збереженої процедури, тепер визначає тип, коли виконується оператор вперше; тип зберігається для всіх наступних викликів збереженої процедури, що містить.

Для параметрів, для яких немає контекстної інформації для визначення типу параметра, сервер припускає, що параметр є символьним рядком із

набором символів за замовчуванням, а не двійковим рядком. Параметри, для яких це неправильно, можуть бути розміщені у виразі CAST ().

Правила, які визначають, як визначаються типи даних про ефективність параметрів та користувацьких змінних, що використовуються в підготовлених операторах, див.

Крім того, аргумент рядків (N) для функцій вікна LAG (), LEAD () та NTILE () тепер повинен бути цілим числом у діапазоні від 1 до 263 включно, в будь-якій з наступних форм:

- беззнаковий цілий константний літерал
- позиційний маркер параметрів (?)
- користувацька змінна
- локальна змінна в збереженій програмі

Крім того, цей аргумент більше не приймає NULL як значення. Для отримання додаткової інформації див. Опис функцій, на які тільки посилаються. (Помилка # 48612, помилка # 99601, помилка # 100150, помилка # 11756670, помилка # 23599127, помилка # 31119132, помилка # 31365678, помилка # 31393719, помилка # 31592822, помилка # 31810577)

Алгоритм сортування файлів тепер підтримує сортування об'єднання в декількох таблицях, а не лише в одній таблиці. (Помилка # 31310238, помилка # 31559978, помилка # 31563876)

При використанні ПРАВИЛЬНОГО СПІЛКУВАННЯ деякі внутрішні об'єкти не перетворювались на ті, що підходять для використання з ЛІВОЮ СПІЛКОЮ за призначенням. Сюди входили деякі списки таблиць, побудованих під час аналізу, але порядок яких не змінювався. Це вимагало підтримання коду для обробки випадків, коли LEFT JOIN спочатку був ПРАВИЛЬНИМ СПИСОКОМ як особливі випадки і був джерелом кількох помилок. Тепер сервер виконує будь-які необхідні розвороти під час аналізу, так що після

синтаксичного аналізу ПРАВИЙ ПРИЄДНАННЯ фактично є, в усіх відношеннях, ЛІВОЮ СПІЛКОЮ. (Помилка # 30887665, помилка # 30964002)

Посилання: Див. Також: Помилка # 12567331, Помилка # 21350125.

Додана підтримка періодичної синхронізації під час запису у файли з операторами `SELECT INTO DUMPFILE` та `SELECT INTO OUTFILE`. Цю функцію можна ввімкнути, встановивши для системної змінної `select_into_disk_sync` значення `ON`; розмір буфера запису `sp` встановлюється за допомогою системної змінної сервера `select_into_buffer_size`; розмір буфера за замовчуванням - 131072 (217) байт. Необов'язкова затримка після синхронізації на диск також може бути встановлена за допомогою системної змінної `select_into_disk_sync_delay`; поведінка за замовчуванням - не допускати будь-якої затримки (тобто часу затримки 0 мілісекунд).

Для отримання додаткової інформації див. Описи системних змінних, на які посилалися раніше.

Дякуємо Facebook за внесок у MySQL 8.0. (Помилка # 30284861)

MySQL тепер реалізує похідне зниження умов для відповідних запитів. Це означає, що для такого запиту, як `SELECT * FROM (SELECT i, j FROM t1) AS dt WHERE i > константа`, тепер у багатьох випадках можна висунути зовнішню умову `WHERE` до похідної таблиці, у цьому випадок, що приводить до `SELECT * FROM (SELECT i, j FROM t1 WHERE i > константа) AS dt`. Раніше, якщо похідна таблиця була матеріалізована, а не об'єднана, MySQL матеріалізувала всю таблицю - в даному випадку `t1` - тоді кваліфікувала рядки умовою `WHERE`.

Коли похідну таблицю неможливо об'єднати із зовнішнім запитом (наприклад, якщо похідна таблиця використовує агрегацію), натискання зовнішньої `WHERE` Умова `RE` до похідної таблиці може зменшити кількість рядків, які потрібно обробити, що повинно покращити продуктивність запиту.

Зовнішню умову `WHERE` можна відсунути безпосередньо до матеріалізованої похідної таблиці, коли похідна таблиця не використовує функції сукупності чи вікна. Крім того, коли похідна таблиця має `GROUP BY` і

не використовує жодних функцій вікна, зовнішня умова WHERE може бути висунута до похідної таблиці як умова HAVING. Якщо в похідній таблиці використовується функція вікна, а зовнішні посилання WHERE посилаються на стовпці, що використовуються в реченні функції вікна PARTITION, умову WHERE також можна відсунути вниз.

Цю оптимізацію не можна застосувати для похідної таблиці, що містить речення UNION або LIMIT.

Щоб увімкнути натискання похідних умов, прапорець `deriva_condition_pushdown` системної змінної `optimizer_switch` (доданий у цьому випуску) має бути увімкнено. Це налаштування за замовчуванням. Якщо цю оптимізацію вимкнено параметром перемикача оптимізатора, ви можете ввімкнути її для певного запиту, використовуючи підказку `DERIVED_CONDITION_PUSHDOWN` (також додану в цьому випуску). Використовуйте підказку оптимізатора `NO_DERIVED_CONDITION_PUSHDOWN`, щоб вимкнути оптимізацію для даного запиту.

Для отримання додаткової інформації та прикладів див. Оптимізацію виведення похідних умов. (Помилка # 59870, Помилка # 88381, Помилка # 11766303, Помилка # 27590273)

Примітки щодо упаковки

Для пакетів RPM та Debian плагіни на стороні клієнта були переміщені до власного пакету клієнт-плагіни. (Помилка # 31584093)

Файл `VERSION` у вихідних дистрибутивах MySQL тепер називається `MYSQL_VERSION` через конфлікт імен із Boost. (Помилка # 31466846)

Для платформ, на яких `systemd` використовується для запуску MySQL, пакунки більше не включають застарілі файли System V: скрипти `mysqld_multi.server` та `mysql.server`, а також man-сторінки `mysql.server.1`, `mysqld_multi.1` та `mysqld_safe.1`. (Помилка # 31450888)

Примітки до схеми продуктивності

Оператор `SHOW PROCESSLIST` надає інформацію про процес, збираючи дані потоку з усіх активних потоків. Однак, оскільки реалізація виконує ітерацію між активними потоками з диспетчера потоків, утримуючи глобальний мьютекс, це має негативні наслідки для продуктивності, особливо на зайнятих системах.

Альтернативна реалізація `SHOW PROCESSLIST` тепер доступна на основі нової таблиці списку процесів схеми продуктивності. Ця реалізація запитує дані активного потоку зі схеми продуктивності, а не диспетчера потоків, і не вимагає мьютекс:

Щоб увімкнути альтернативну реалізацію, увімкніть системну змінну `performance_schema_show_processlist`.

Альтернативна реалізація `SHOW PROCESSLIST` також застосовується до команди `mysqladmin processlist`.

Альтернативна реалізація не застосовується до таблиці `INFORMATION_SCHEMA PROCESSLIST` або команди `COM_PROCESS_INFO` протоколу клієнта / сервера MySQL.

Щоб гарантувати, що реалізації за замовчуванням та альтернативні варіанти дають однакову інформацію, мають бути дотримані певні вимоги до конфігурації; див. таблицю списку процесів.

Інтерфейс SQL до останніх подій, записаних до журналу помилок сервера MySQL, тепер доступний за допомогою запитів у новій таблиці помилок журналу ефективності. Цей стіл має фіксований розмір, при цьому старі події автоматично відкидаються за необхідності, щоб звільнити місце для нових.

Таблиця заповнюється, якщо конфігурація журналу помилок включає компонент журналу журналу, який підтримує цю можливість (на даний момент традиційні формати `log_sink_internal` та JSON-формат `log_sink_json`). Кілька нових змінних стану надають інформацію про роботу таблиці `error_log`. Див. Таблицю журналу помилок.

Підключається аутентифікація

Ці зміни були внесені для плагінів автентифікації LDAP:

Для плагіна автентифікації LDAP SASL метод аутентифікації SCRAM-SHA-1 не підтримується в системах SLES 12 та 15 та EL6. Методом за замовчуванням у цих системах тепер є GSSAPI.

Якщо хост LDAP не встановлений, пул підключення LDAP не ініціалізується, що дозволяє успішно встановити плагін автентифікації у випадках, коли раніше він не вдався. (Це може бути у випадку, коли сайт спочатку встановлює плагін, а потім налаштовує його пізніше.)

Якщо параметр з'єднання LDAP змінюється під час виконання, пул з'єднань LDAP повторно ініціалізується для першої наступної спроби автентифікації.

Якщо сервер LDAP перезапущений, існуючі підключення в пулі підключень стають недійсними. Плагін автентифікації LDAP виявляє цей випадок та повторно ініціалізує пул підключень, і (для плагіна LDAP SASL) завдання SASL повторно надіслано.

(Помилка # 31664270, помилка # 31219323)

Примітки щодо синтаксису SQL

Тепер парсер підтримує вирази запитів у дужках, використовуючи такий синтаксис:

```
1( query_expression )  
2 [order_by_clause]  
3 [limit_clause]
```

4 [into_clause]

Можливі й інші варіації; див. вирази запитів у дужках (помилка #30592703)

Тепер можна передавати значення інших типів на РІК, використовуючи або функцію CAST (), або функцію CONVERT (). Тепер ці функції підтримують РІЧНІ значення однієї або двох цифр у діапазоні 0-99 та чотиризначні значення в діапазоні 1901-2155. Ціле число 0 перетворюється на рік 0; рядок, що складається з одного або декількох нулів (після можливого усічення), перетворюється на 2000 рік. Відливання додає 2000 до значень в діапазоні 1-69 включно та 1900 до значень в діапазоні 70-99 включно.

Рядки, що починаються з однієї, двох або чотирьох цифр, за якими слідує принаймні один нецифровий символ (і, можливо, інші цифрові або нецифрові символи), усікаються перед перетворенням на YEAR; у таких випадках сервер видає попередження про скорочення. Значення з плаваючою комою округлюються до перетворення; CAST (1944,5 ЯК РІК) повертає 1945 через округлення, а CAST ("1944,5" ЯК РІК) повертає 1944 (із попередженням) через усічення.

DATE, DATETIME і TIMESTAMP додаються до РОЧНОЇ частини значення. Значення TIME визначається для поточного року. Якщо не вказати значення, яке буде передано як значення ЧАСУ, це може дати інший результат від очікуваного; CAST ("13:47" AS YEAR) повертає 2013 рік через усічення значення рядка, а CAST (TIME "13:47" AS YEAR) повертає 2020 рік станом на рік цього випуску.

Передача значень GEOMETRY до YEAR не підтримується. Привід несумісного типу або поза діапазоном або нелегальне значення повертає NULL.

YEAR також може бути використаний як тип повернення для функції JSON_VALUE (). Ця функція підтримує лише чотиризначні роки, а в іншому випадку дотримується тих самих правил, що застосовуються до CAST () та CONVERT () при виконанні кастів до YEAR.

Для отримання додаткової інформації див. Опис функції CONVERT ().

При виборі значення стовпця TIMESTAMP тепер можна перетворити його із системного часового поясу на UTC DATETIME під час його отримання, використовуючи оператор AT TIME ZONE, який реалізовано для функції CAST () у цьому випуску.

Синтаксис CAST (значення AT TIME ZONE AS AS DATETIME [(precision)]), де значенням є TIMESTAMP, а специфікатором є один із [INTERVAL] '+00: 00' або 'UTC'. (INTERVAL не є обов'язковим для першої форми специфікатора і не може використовуватися з "UTC".) Точність значення DATETIME, що повертається приводом, за бажанням може бути вказана до 6 знаків після коми.

Також підтримуються значення, які були вставлені в таблицю із зміщенням часового поясу.

ЧАСОВУ ЗОНУ не можна використовувати з CONVERT () або в будь-якому іншому контексті, крім як частина виклику функції CAST (). Ключове слово ARRAY та створення багатозначних індексів також не підтримуються під час використання ЗОНИ ВЧАСУ.

Короткий приклад наведено тут:

```
mysql> SELECT @@system_time_zone;
```

```
+-----+  
| @@system_time_zone |
```

```
+-----+  
| EDT |
```

```
+-----+  
1 row in set (0.00 sec)
```

```
mysql> CREATE TABLE ex (ts TIMESTAMP);
```

```
Query OK, 0 rows affected (0.81 sec)
```

```
mysql> INSERT INTO ex VALUES
```

```
> ROW(CURRENT_TIMESTAMP),
```

```
> ROW('2020-07-31 21:44:30-08:00');
```

```
Query OK, 2 rows affected (0.09 sec)
```

Records: 2 Duplicates: 0 Warnings: 0

```
mysql> TABLE ex;
```

```
+-----+
| ts      |
+-----+
| 2020-07-28 21:39:31 |
| 2020-08-01 01:44:30 |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT ts, CAST(ts AT TIME ZONE 'UTC' AS DATETIME) AS ut
FROM ex;
```

```
+-----+-----+
| ts      | ut      |
+-----+-----+
| 2020-07-28 21:39:31 | 2020-07-29 01:39:31 |
| 2020-08-01 01:44:30 | 2020-08-01 05:44:30 |
+-----+-----+
2 rows in set (0.00 sec)
```

Примітки до плагіна X

У конкретних умовах припинення підключення до протоколу X може спричинити несподівану зупинку сервера MySQL. (Помилка # 31671503)

Додана або змінена функціональність

Покращено перевірку привілеїв LOCK TABLES для переглядів. (Помилка # 31304432)

Ви можете використовувати новий механізм відмови від асинхронного підключення MySQL Server для автоматичного встановлення асинхронного (джерела до репліки) підключення реплікації до нового джерела після того, як існуюче з'єднання з репліки до його джерела не вдається. Помилка з'єднання, якщо потік вводу-виводу реплікації зупиняється через зупинку джерела або через збій мережі. Механізм асинхронного переключення підключення може використовуватися для синхронізації репліки з декількома серверами MySQL або групами серверів, які діляться даними. Щоб активувати перехід на асинхронне підключення для каналу реплікації, встановіть SOURCE_CONNECTION_AUTO_FAILOVER = 1 у операторі CHANGE

MASTER TO для каналу та встановить список джерел для каналу, використовуючи `asynchronous_connection_failover_add_source` і `asynchronous_connection_failover_Detect_failover_Detect_failover`

Нова змінна `innodb_extend_and_initialize` дозволяє налаштувати, як InnoDB розподіляє простір для файлів на таблицю та загальних табличних просторів у Linux. За замовчуванням, коли операція вимагає додаткового простору в табличній області, InnoDB виділяє сторінки в табличну область і фізично записує NULL на ці сторінки. Ця поведінка впливає на продуктивність, якщо нові сторінки часто виділяються. Починаючи з MySQL 8.0.22, ви можете вимкнути `innodb_extend_and_initialize` в системах Linux, щоб уникнути фізичного запису NULL на щойно виділені сторінки табличного простору. Коли `innodb_extend_and_initialize` вимкнено, простір виділяється за допомогою викликів `posix_fallocate ()`, які резервують простір, не записуючи фізично NULL.

Операція `posix_fallocate ()` не є атомною, що робить можливим помилку між виділенням простору у файл табличної області та оновленням метаданих файлу. Така помилка може залишити щойно виділені сторінки в неініціалізованому стані, що призведе до помилки при спробі InnoDB отримати доступ до цих сторінок. Щоб запобігти цьому сценарію, InnoDB пише запис журналу повтору перед виділенням нової сторінки табличного простору. Якщо операція розподілу сторінки переривається, вона відтворюється із запису журналу повторення під час відновлення.

Щоб дозволити паралельні операції DML і DDL в таблицях довідок MySQL, операції читання, які раніше отримували блокування рядків у таблицях доручень MySQL, тепер виконуються як незаблоковані читання. Операції, які тепер виконуються як неблокуючі зчитування в таблицях доручень MySQL, включають:

Оператори SELECT та інші оператори лише для читання, які зчитують дані з таблиць надання через списки об'єднань та підзапити, включаючи оператори SELECT ... FOR SHARE, використовуючи будь-який рівень ізоляції транзакцій.

Операції DML, які зчитують дані з таблиць надання (через списки об'єднань або підзапити), але не змінюють їх, використовуючи будь-який рівень ізоляції транзакцій.

У висловлюваннях, які більше не отримують блокування рядків під час зчитування даних із таблиць надання, з'являється попередження, якщо воно виконується під час використання реплікації на основі операторів.

При використанні `-binlog_format = змішаних`, операції DML, які зчитують дані з таблиць надання, тепер записуються в двійковий журнал як події рядків, щоб зробити операції безпечними для змішаного режиму реплікації.

Оператори SELECT ... FOR SHARE, які зчитують дані з таблиць надання, тепер подають попередження. З пропозицією FOR SHARE блокування читання не підтримується в таблицях доручень.

Операції DML, які зчитують дані з таблиць доручень і виконуються за допомогою рівня ізоляції `SERIALIZABLE`, тепер повідомляють про попередження. Блокування читання, які зазвичай отримуються під час використання рівня ізоляції `SERIALIZABLE`, не підтримуються таблицями надання.

З MySQL 8.0.22 системна змінна `group_replication_ip_whitelist` застаріла, а на зміну додано системну змінну `group_replication_ip_allowlist`. Системна змінна працює так само, як і раніше, лише термінологія змінена.

Для обох системних змінних значенням за замовчуванням є `AUTOMATIC`. Якщо для однієї із системних змінних встановлено значення, визначене користувачем, а для іншої немає, використовується змінене значення. Якщо для

обох системних змінних встановлено користувачське значення, використовується значення `group_replication_ip_allowlist`.

З MySQL 8.0.22, інструкції `START SLAVE`, `STOP SLAVE`, `SHOW SLAVE STATUS`, `SHOW SLAVE HOSTS` і `RESET SLAVE` застаріли. Натомість слід використовувати такі псевдоніми:

Замість `START SLAVE` використовуйте `START REPLICA`

Замість `STOP SLAVE` використовуйте `STOP REPLICA`

Замість `SHOW SLEVE STATUS` використовуйте `SHOW REPLICA STATUS`

Замість `SHOW HAVES HOSTS` використовуйте `SHOW REPLICAS`

Замість `RESET SLAVE` використовуйте `RESET REPLICA`

Висловлювання працюють так само, як і раніше, змінилася лише термінологія, яка використовується для кожного висловлення та його результатів.

Нові змінні стану були додані як псевдоніми відповідних змінних стану. І стара, і нова версії операторів оновлюють як стару, так і нову версії цих змінних стану:

`Com_slave_start` еквівалентно `Com_replica_start`

`Com_slave_stop` – еквівалент

алент до `Com_replica_stop`

`Com_show_slave_status` еквівалентно `Com_show_replica_status`

`Com_show_slave_hosts` еквівалентно `Com_show_replicas`

Оператор `ALTER DATABASE` тепер підтримує опцію `READ ONLY`, яка контролює, чи дозволяти модифікацію бази даних та об'єктів у ній. Цей параметр корисний для міграції бази даних, оскільки базу даних, для якої увімкнено **ТІЛЬКИ ЧИТАННЯ**, можна перенести на інший екземпляр MySQL без побоювань, що базу даних можна змінити під час операції. Див. **ЗВІТ ЗМІНИ БАЗИ ДАНИХ**

Нова таблиця INFORMATION_SCHEMA з назвою SCHEMATA_EXTENSIONS відображає параметри бази даних. Наразі для баз даних, доступних лише для читання, відображається ТІЛЬКИ ЧИТАННЯ = 1. Див. Таблицю INFORMATION_SCHEMA SCHEMATA_EXTENSIONS.

Виправлено помилки

InnoDB: Вилучено код, пов'язаний із підтримкою транзакцій для вибірки гістограм, включаючи пов'язаний код твердження, що спричинив невдалі тести. Для вибірки гістограми не потрібна підтримка транзакції. (Помилка # 31787736)

InnoDB: Інформація шифрування не була встановлена для операцій запису журналу повторення журналу архіву журналу. (Помилка # 31690196)

InnoDB: Функція TTASEventMutex :: exit була оптимізована для ARM64. Дякуємо Круналу Баускару за внесок. (Помилка # 31589019, помилка # 100132)

InnoDB: InnoDB не вдалося скомпілювати з увімкненою опцією DISABLE_PSI_RWLOCK CMake. (Помилка # 31578289)

InnoDB: Рівень ізоляції транзакції, для якого встановлено значення READ UNCOMMITTED для вибірки гістограм, щоб уникнути непотрібного пошуку старих версій записів, не був скинутий після завершення операції вибірки. (Помилка # 31564407)

InnoDB: Запит, який оновив кластерний індекс внутрішньої тимчасової таблиці, повернув неправильний результат. Модифіковані сторінки кластерного індексу не були додані до списку очищення, що призвело до втрачених змін, коли змінені сторінки були виселені з пулу буферів. (Помилка # 31560679)

Посилання: Це питання є регресією: Помилка # 29207450.

InnoDB: Видалено залежність побудови від бібліотеки Boost, визначеної для механізму зберігання TempTable. (Помилка # 31505048)

InnoDB: Було реалізовано обхідний спосіб вирішення проблеми компілятора Clang у 32-розрядних збірках, що призводить до того, що значення

ATOMIC_LLONG_LOCK_FREE визначається як «іноді без блокування», тоді як __atomic_always_lock_free повертає true для того самого типу на тій же платформі. (Помилка # 31504609)

InnoDB: Таблиця формату рядків REDUNDANT, створена в попередній версії MySQL, де формат рядків не був чітко визначений, дозволяла додавати індекс, який перевищував обмеження розміру стовпця індексу формату рядків. (Помилка # 31479542, помилка # 99791)

InnoDB: Помилка операції DML у стовпці, визначеному багатозначним індексом. (Помилка # 31479282)

InnoDB: Помилка сталася під час обертання головного ключа. Об'єкт пам'яті скасування табличного простору було звільнено передчасно. (Помилка # 31467626)

InnoDB: Невикористаний фізичний код читання вперед вилучено з інтерфейсу паралельного читання. (Помилка # 31429385)

InnoDB: Операція обертання головного ключа не змогла пропустити скасовану табличну область, яка вже була усічена, що призвело до помилки твердження під час вимкнення сервера. (Помилка # 31400195)

InnoDB: Після імпорту простору таблиць для стиснутої сторінки таблиці сторінки більше не стискались, а метадані INFORMATION_SCHEMA.INNODB_TABLESPACES неправильно вказували на стискання сторінок. Інформація про стиснення таблиці була недоступною під час операції імпорту. (Помилка # 31396947)

InnoDB: операція відкоту та оновлення після виконання миттєвої операції DDL викликала твердження. (Помилка # 31391126)

InnoDB: Заблокована система читання-запису в системі журналу (log_sys) спричинила регресію продуктивності робочих навантажень, пов'язаних з процесором. (Помилка # 31389135)

InnoDB: Компіляція з увімкненим параметром UNIV_ENCRYPT_DEBUG спричинила помилки компіляції. (Помилка # 31369540)

InnoDB: DDL-операції з розділеною таблицею можуть спричинити помилку. TABLE_SHARE та об'єкти екземпляру таблиці були відкриті для всіх розділів без потреби. (Помилка # 31365127)

InnoDB: Після зміни сортування стовпців VARCHAR з utf8mb4 на utf8mb4_bin в операції ALTER TABLE на місці та додавання індексу до того самого стовпця, запит з урахуванням регістру в стовпці VARCHAR повернув неправильний результат. Зіставлення стовпців VARCHAR було змінено в словнику даних, але не в об'єкті таблиці в пам'яті. Отже, індекс, створений у стовпці VARCHAR, використовував застарілу інформацію стовпця, що викликало порівняння для використання раніше визначеного порівняння. (Помилка # 31361838)

Висновки

Головною ціллю проекту є впровадження технології QR та використання її для ідентифікації та аутентифікації користувачів. Під час створення тесту викладач ,використовуючи основні модклі програми може з легкістб створювати унікальні QR коди в один клік . Кожен QR код є унікальним та має свій час існування , після закінчення якого QR код видаляється.

Після сканування в базі даних встановлюється час видалення QR коду як

Під час розробки додатку було використано низку новітніх програм та технологій. Беручи до уваги список використаного програмного забезпечення можна зробити наступні висновки:

1. Додаток розроблено та розгорнуто відповідно до сучасних стандартів програмування.
2. Програмний код зашифровано та мінімізовано за допомогою допоміжних модулів Angular 8.
3. Всі модулі програми винесено в окремі модулі та компоненти , що робить програму масштабованою та легкою для підтримки та додавання нового програмного коду.

4. Всі данні про користувача та вся інформація зберігається на сервері Azure у зашифрованому вигляді ,що забезпечує конфіденційність та цілісність даних користувачів.
5. Програма є автономною , що є значним плюсом та забезпечить неперервність та роботу без збоїв.
6. Програма легка у використанні та не потребує особливих навичок , завдяки вдалому поєднанню front-end та back-end частин потужностями Angular 6.
7. Всі данні легко збираються і оброблюються , що дає змогу отримати високу швидкість додатку.
8. Додаток знаходиться на хмарній платформі Azure , що свідчить про високу доступність та цілісність додатку , а також високу швидкість та безперебійність в роботі.

РОЗДІЛ 3

ОГЛЯД АНАЛОГІВ ТА ПОРНІВНЯННЯ ІЗ ДОДАТКОМ BRAIN BENEFITS

Без тестів сьогодні не обходиться жоден викладач, будь то шкільний учитель чи професор багатотисячного онлайн-курсу. Вибір варіанту з декількох - не єдина можливість: в опитування можна вставляти картинки і відео, змінювати формати завдання майже до безкінечності. Було обрано сім сервісів, які також надають можливість створення тестів.

3.1 Аналіз конкурентів

LEARNINGAPPS

Сервіс для підтримки навчального процесу шляхом створення та збереження інтерактивних вправ ігрового характеру. Орієнтований на школярів та дорослих. Завдання найкраще підходять для проведення узагальнення та систематизації знань з певної дисципліни. Вправи, або блоки завдань, не замінюють повноцінний урок, вони лише можуть бути складовою частиною певного виду діяльності під час уроку. Даний сайт буде цікавий перш за все школярам молодших та середніх класів, оскільки вправи побудовані у вигляді незвичних інтелектуальних ігор. Проте є можливість вибрати рівень складності завдань – від дошкільного до післядипломної освіти. За допомогою платформи можна створювати 18 різновидів завдань: знайти пару, кросворд, класифікація, числова пряма, просте упорядкування, фрагменти зображення, вільна текстова відповідь, вікторина, заповнити пропуски, колекція вправ, аудіо-, відеоконтент, перший мільйон, пазл «Вгадай слово», шибениця, знайти слова, гра «Парочки». Категорія предметів найрізноманітніша: англійська, німецька, французька, іспанська, італійська, українська, російська та інші мови, інформатика, історія, інструменти, астрономія, біологія, географія, економіка, математика, мистецтво, музика, здоров'я, навколишній світ, політологія, релігієзнавство, професійна освіта, філософія, психологія, технічні науки, трудове навчання,

фізкультура, фізика і хімія. Пропонується широкий вибір мов інтерфейсу, в тому числі українська. Учителю потрібно створити обліковий запис, вказавши свій логін/email і пароль. Учням також варто зареєструватися на сайті за допомогою електронної скриньки, тоді учитель зможе додати їх у свій клас. Посилання на завдання можна надсилати у вигляді QR-коду, який система автоматично створює до кожного завдання. Також посилання на вправу можна вбудувати у свій сайт. Учитель чи репетитор може обирати запропоновані сайтом предметні категорії та варіанти завдань у розділі «Перегляд вправ». Детальніше:

МАЙСТЕР-ТЕСТ

Безкоштовний освітній сервіс, орієнтований на створення тестів та проведення онлайн тестування з навчальною метою. Чудово підходить для організації тематичного та контрольного оцінювання знань. Інтерфейс ресурсу представлений українською, англійською, російською мовами. Процес реєстрації надзвичайно простий – вписуєте ім'я, прізвище, електронну пошту та пароль, далі підтверджуєте реєстрацію, вказуєте роль (викладач чи студент) – і можете розпочинати створення першого тесту. Натискаєте на «Створити перший тест зараз» або у розділі «Мої тести» на «Створити новий тест». У новому вікні вгорі записуємо текст запитання та за потреби додаткову роз'яснювальну інформацію. Далі обираємо тип запитання, а саме: одна правильна відповідь, кілька відповідей, слово, номер чи співвідношення. Кількість рядків з варіантами відповідей можна збільшувати, натиснувши на «Додати відповідь». «Вага питання» означає кількість балів за одне запитання. У віконечках навпроти варіантів за бажанням позначаємо галочкою правильну відповідь. В кінці натискаємо «Готово» та переходимо до наступного запитання.

Google форми

Google Форми - частина офісного інструментарію Google Drive. Мабуть, це один з найшвидших і простих способів створити своє опитування або тест:

пишемо завдання, вибираємо тип відповіді (вибір з кількох варіантів, написання власного) - готово! Одержаний тест можна відправити студентам по електронній пошті або вбудувати на свій сайт за допомогою спеціального коду. Для прискорення роботи рекомендовано додати плагін Flubaroo - він автоматично перевіряє відповіді учнів і ставить оцінки за заданими критеріями. Форми абсолютно безкоштовні - для використання ресурсу потрібно тільки мати акаунт Google.

У використанні форми досить прості та не потребують спеціальних навичок або ж додаткових коштів, що робить продукт досить привабливим для використання. Оптимальний варіант, якщо потрібно створити тести за лічені хвилини та розіслати. Форми не є порталом а лише механізмом для створення тестів, що виключає наступне ПО зі списку прямих конкурентів.

Quizlet

За допомогою Quizlet можна створювати тести, де учні будуть вибирати вірні відповіді із запропонованих, зіставляти зображення і інформацію або вписувати власні варіанти. Користуватися дуже легко - в інтерфейсі можна розібратися буквально за пару хвилин, до того ж, Quizlet підтримує російську мову і працює на Android і iOS. Quizlet безкоштовний, але буває і підписка Plus за 10\$ - вона дозволить завантажувати власні картинки і створювати необмежену кількість навчальних груп.

Досить легкий у використанні, але потребує завантаження додаткового програмного забезпечення та мінімальних навичок володіння програмою. Для створення додаткових тестів потрібно купувати підписку. В цілому програма досить непогана, але не містить в собі стимулів: відсутня система нарахування бонусів за проходження тестів та за відвідування порталу напротязі тижня, що негативно впливає на потік кадрів. Програма не має на меті підвищення рівня школярів та студентів, а є лише сервісом для надання послуги створення та проходження тестів, що в свою чергу виключає Quizlet із списку конкурентів.

Proprofs

Proprofs готує тести на будь-який смак - можна запропонувати на вибір один або кілька варіантів, попросити заповнити пропущене слово або написати розгорнуту відповідь. Сервіс дозволяє вставляти в завдання текстові документи та презентації, файли PDF, а також зображення, аудіо- та відеофайли. Завершивши роботу над тестом, можна залишити його в загальному доступі на сайті Proprofs або вбудувати на свою сторінку.

Хоча сервіс безкоштовний, можливості Proprofs розширюються в платних тарифах. Викладачам пропонують звернути увагу на тарифи Basic і Professional. Перший відкриває всі основні можливості сервісу і дозволяє створювати необмежену кількість тестів за \$ 20 в місяць; другий дозволяє об'єднувати студентів в закриті групи і обійдеться в \$ 40. Нові користувачі можуть безкоштовно користуватися всіма можливостями Proprofs протягом 15 днів після реєстрації.

Досить привабливий продукт із можливістю створення тестів, використовуючи аудіо, відео, та текстове подання інформації для тестів, що в свою чергу є родзинкою даного порталу. В даній програмі є можливість не лише створення, а й поширення власних тестів не лише серед своєї аудиторії, а й серед інших користувачів порталу. Беручи до уваги всі ці фактори, можна зробити висновок, що **Proprofs** – досить непоганий конкурент програмі **Brain Benefits**. Та зіставляючи ці дві програми, досить швидко можна зрозуміти, що **Brain Benefits** надає можливості для розвитку, шляхом стимулювання та заохочення студентів, має величезну базу тестів та викладачів та унікальні алгоритми для розрахунку балів за відвідування та проходження тестів різної складності.

Kahoot

Kahoot! дозволяє подавати в форматі опитувань і тестів мало не весь навчальний матеріал. Щоб налагодити зворотний зв'язок з учнями, можна обіграти нові теми у формі простих питань і відповідей, а закріпити знання за

допомогою більш докладного тестування. Kahoot! розрахований на застосування в класі - викладач показує матеріал на головному екрані, а в цей час школярі відповідають на питання і обговорюють інформацію, використовуючи спеціальний клієнт для комп'ютерів або браузер на смартфонах (Android, iOS, Windows Phone). Для того щоб увійти в віртуальну класну кімнату, учні повинні ввести спеціальний код, який надішле викладач.

Досить потужний портал для створення та проходження тестів, особливо цікавим є створення та розповсюдження унікального коду для доступу до конкретного тесту. Та Kahoot розроблено для створення та проходження тестів в ріал-тайм часі. Також портал слугую для лише для комфортної роботи із тестами, але ввілсутня система заохочення та додаткових стимулів, що робить із Kahoot досить непоганий портал в якого є чому повчитись, але виключає його з прямих конкурентів **Brain Benefits**.

Easy test maker

Найцікавіше в Easy Test Maker - можливість створювати завдання, де потрібно вибирати вірні і помилкові твердження. Тексти можна відформатувати для зручності читання на планшеті або скачати в форматах .pdf або .doc, щоб провести тестування в більш традиційному форматі. Сервіс вміє перемішувати питання і варіанти відповідей, щоб учням було складніше списувати.

Висновки

Проводячи оцінку аналогів та беручи до у ваги їх слабкі та сильні сторони, можна дійти висновку, що більшість програм лише надають змогу створювати та проходити тести. Більшість програм сфокусовані на покращенні візуальної частини та поліпшенні роботи із програмою. Але повторне проходження тесту з іншого аккаунту залишається, списування та використання чужий результатів також залишаються не вирішеним питанням для основних конкурентів платформи.

Виходячи з вищезазначеного можна з упевненістю сказати , що додаток не має аналогів та є унікальним порталом , що надає змогу не лише створювати та проходити тести . Додаток стимулює розвиток та розумову діяльність користувачів. Такий результат досягається за рахунок стимулювання та заохочення студентів до навчання шляхом надання їм певних винагороджень на певних етапах та за сумою отриманих балів.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП-ПРОЕКТУ

В цьому розділі буде проведено аналіз стартап проекту «Програмне забезпечення для проведення незалежного on-line тестування з використанням qr кодів».

4.1. Опис ідеї проекту

Метою стартапу є розробка онлайн платформи для створення та впровадження онлайн тестування з використанням технології QR.

У таблиці 4.1 зображено зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1. Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення оптимальних умов для проходження незалежного тестування за допомогою QR технології	Навчальна сфера	Підвищення надійності сесії користувача
	Найм персоналу	Прозорі умови створення та генерації унікальних тестів.

Проект створений для тестування в різних цілях. Унікальність проекту в комплексності рішення. Багато функцій можна знайти в інших проектах, але завдання проекту створити єдину екосистему: від створення тестів / опитувань, управління контактами, публікацій тестів, електронної платформи для сертифікації до адаптивного інструменту персонального розвитку . Кожен автор тесту може згенерувати кінцеву кількість унікальних QR кодів для учасників тестування. Будь-який користувач може створити тест для своїх цілей. Аналогічно ресурс дозволяє створювати вакансії будь-якого напрямку.

4.2 Опис основних функцій проекту

З точки зору викладачів та роботодавців проект є досить зручним інструментом для створення впровадження та контролю процесу тестування.

Кожен викладач має змогу внести перелік питань з дисципліни ,рівень залишкових знань з якої він би хотів перевірити. Після внесення питань , викладач вносить список студентів,їх поштові скриньки та їх групу – на цьому етапі відбувається формування унікальних QR кодів для кожного учасника тестування. Після створення кодів вони відправляються кожному учаснику тестування як додаток до листа із основною інформацією про тест.

Завдання учасника тестування лише відсканувати QR код який автоматично відкриє необхідну сесію з унікальним варіантом . Після сканування QR коду починається відлік часу та по закінченню QR код видаляється і стає лише малюнком, який не несе ніякого інформаційного навантаження.

Після проходження тестів усіма учасниками результати обробляються та висилаються викладачеві.

Всі тести , що відносяться до відомих дисциплін зберігаються та можуть бути використані іншими викладачами для створення власних.

4.3 Бізнес модель проекту

Заслуговує на увагу бізнес-модель проекту, яка побудована на диференціації джерел монетизації і включає в себе наступні складові:

- CPA (cost per acquisition). Дає можливість навчальним закладам і учням знайти один одного. У підсумку, користувачі отримують пільгові умови навчання, а навчальний заклад студента.
- Преміум-акаунти. Вся поточна функціональність залишиться безкоштовною, але плануються вузькоспеціалізовані функції, які будуть доступні тільки преміум-користувачам (наприклад, видача сертифікатів).

- Market place для тестів. Надання можливості створення експертами бази тестових завдань для комерційного використання. It's quiz в свою чергу буде стягувати невелику комісію з внутрішніх транзакцій.

Проект знаходиться на самому початку свого розвитку. Його ідея з'явилася всього чотири місяці тому і змогла в найкоротші терміни перерости в працюючий портал тестування знань і визначення рівня компетенцій.

4.4 Аналіз ринкових можливостей запуску стартап проекту

тестування - це різносторонній інструмент, який можна використовувати не тільки для контролю рівня знань, але і в наймі персоналу, вивчення нової інформації з метою розвитку. Ідея ресурсу з'явилася з особистого досвіду розробників в сфері реалізації проектів в галузі освіти і підбору персоналу в сфері ІТ

Рентабельність - це показник економічної ефективності використання ресурсів або інвестицій. Він показує вдалося чи не вдалося повернути вкладення і на скільки. Судячи з даних таблиці ринок є привабливим для входження.

Заслуговує на увагу бізнес-модель проекту, яка побудована на диференціації джерел монетизації і включає в себе наступні складові

Надалі визначаємо потенційні групи клієнтів:

Таблиця 4.5.

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Незалежність	навчання	Простота у використанні.	Максимальна незалежність та прозорість формування тесту

2	Термін мережі та споживання енергії	Найм на роботу	Масштабування проекту	Можливість швидкого внесення нових питань
---	-------------------------------------	----------------	-----------------------	---

У зв'язку з тим, що аудиторія та сфера в цілому обширна, тенденції дуже часто и швидко змінюються і потрібно швидко адаптуватися під них, потенційні клієнти також це розуміють, та намагаються впроваджувати у проекти нові та актуальні методи.

При застосуванні даної технології існують певні загрози. (таблиця 4.6).

Таблиця 4.6.

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Попиту	Спроба підробити QR код	Виявлення слабких полів бази даних та виключення наступних з інформаційного навантаження QR коду
2.	Економічна	Потреба більш швидких серверів	Розміщення реклами на сторінках додатку та отримання прибутку за перегляди
3.	Конкуренція	Можливість створення аналогів від більш відомих компаній	Продаж ідей та алгоритмів
4.	Науково-технічна	Швидкий розвиток науки	Моніторинг наукових новин та пошук нових шляхів вдосконалення проекту

Ризики існують, тож потрібно мати міцний фундамент у вигляді документів, сертифікатів, які підтверджують усі можливі наміри, результати тестувань та виділення основних переваг цього алгоритму незалежного тестування.

Аналіз підтвержує, що наш проект все ще потребує значних зусиль для того, щоб увійти у ринок, зафіксуватися та пропонувати свої можливості своїм клієнтам.

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі.

На основі проведеного аналізу можна сформулювати перелік основних факторів, які свідчать про високу конкурентоспроможність проекту (табл. 4.10).

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
	Ціна	Додаток є безкоштовним.
	Актуальність	В час пандемії проект є дуже актуальним.
	Попит	Активна тенденція та розвиток ринку свідчить, що попит є
	Гнучкість	Є можливість подальшої модифікації та реалізації для будь-яких завдань пов'язаних з поліпшенням незалежності оточуючих факторів в роботі алгоритма
	Інноваційність	Робить українську науку на рівні з іншими країнами.

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11). Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін

Фактор	Рейтинг товарів-конкурентів у
--------	-------------------------------

п/п	конкурентоспроможності	бали 1-20	порівняння з проектом						
			3	2	1		1	2	3
	Ціна	8							
	Актуальність	7							
	Попит	0							
	Гнучкість	7							
	Іноваційність	8							

З таблиць 4.10 та 4.11 що фактори конкурентноспроможності дозволять вийти на ринок, але успіху можна буде досягти лише за рахунок якісної реалізації та продуманої маркетингової політики.

Таблиця 4.12. SWOT- аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> 1. Інноваційність; 2. Вартість. 3. Гнучкість 	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> 1) Відсутність довіри; 2) Велика конкуренція 3) Велика витрата ресурсів до самих продажів на рекламу
---	---

Можливості: 1. Збільшення продаж; 2. Розширення ринку за рахунок іноземних замовників; 3. Отримання тендерів на послуги.	Загрози: — Цінова конкуренція в зв'язку з появою нових гравців на ринку. — За рахунок великої тенденції і конкуренції - технологічна конкуренція
--	---

Це знову підтверджує, що навіть незважаючи на свою специфіку, наш проект потребує значних зусиль для того, щоб увійти у ринок, зафіксуватися та пропонувати свої можливості своїм клієнтам

На основі SWOT-аналізу розробляємо альтернативи ринкової.

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Стратегія нейтралізації ринкових загроз сильними сторонами стартапу	70%	3 місяці
2	Стратегія компенсації слабких сторін стартапу наявними ринковими можливостями	70%	3 місяці
3	Стратегія виходу з ринку	80%	6 місяців

З зазначених альтернатив обираємо стратегію компенсації слабких сторін стартапу наявними ринковими можливостями.

4.5 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
-------	-----------------------------	---	------------------------------------	--------------------------------------	--------------------------

	потенційних клієнтів		групи (сегменту)		
1	навчання	Так	Високий	Середня	Середня
2	Підбір персоналу	Так	Високий	Висока	Середня

Для роботи з вибраним цільовими групами користувачів ринку необхідно сформувати базову стратегію розвитку.

Таблиця 4.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Підсилення сильних сторін стартапу за рахунок ринкових можливостей	Перемовини з компаніями, які представляють цільові групи потенційних клієнтів	Виокремлення переваг цього способу у грошовому еквіваленті для майбутніх споживачів.	Стратегія підкріплення своїх переваг

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Забирати існуючих	Так алгоритм підбору тестів	Стратегія підкріплення своїх переваг

На основі вимог споживачів з обраного сегменту до постачальника і продукту, а також в залежності від стратегії розвитку та стратегії конкурентної поведінки розробляємо стратегію позиціонування яка визначається у формування ринкової позиції, за яким споживачі мають ідентифікувати проект.

Таблиця 4.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Цілкова підтримка на етапі інтеграції	Відкритість до вирішення питань	Обізнаність свого продукту, допомога в інтеграції. Формування лояльності і прихильності споживачів, підтримка вхідних бар'єрів.	Гнучкість. Термін служби. Енергоефективність..

Результатом даного підрозділу є система рішень щодо ринкової поведінки компанії, вона визначає в якому напрямі буде працювати компанія на ринку

4.6 Розроблення маркетингової програми стартап-проекту

Під час розроблення маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач. У таблиці 4.18 підсумовуємо результати аналізу конкурентоспроможності товару.

Таблиця 4.18. Визначення ключових переваг концепції товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Конкурентоспроможності	Унікальність, простота	Легка інтеграція методу та модифікація під потреби

Висновки

Платформа - проект, що володіє великим потенціалом і доводить, що втілити ідею можна швидко і ефективно, добре вивчивши клієнтів і сфокусувавшись на цілях.

Команда постійно шукає нові шляхи розвитку проекту та нових партнерів, а також запрошує всіх оцінити проект і прийняти участь в його розвитку.

Основний упор у розвитку проекту зроблений на автоматизацію процесу оцінки знань, його інтеграцію в роботу підприємств. Зараз планується співпраця платформи з іншими масштабними українськими проектами.

Результатом проекту є стратегія виходу на ринок, маркетинговий план та обрані стратегії співпраці з майбутніми(потенційними) клієнтами. За рахунок аналізу також було виявлено, що впровадження ідеї з технологією QR потенційним клієнтам є реальна можливість, але для цього потрібно проводити тестування, дані, можливість інтеграції нашого методу в проект клієнтів і так далі.

Галузь має дуже позитивну тенденцію та розвиток у всьому світі, це вказує на те, що якщо правильно підійти до виходу на ринок, свій обсяг клієнтів обов'язково отримаємо.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

В даній було проведено повний опис програмного забезпечення, що використовувалось для створення онлайн платформи для тестування. Описано основне програмне забезпечення для проведення , зберігання та видалення тестів із бази за допомогою QR.

На першому етапі було проведено дослідження серед студентів, школярів та їх батьків у два етапи:

- Аналіз готовності учнів та студентів до сприйняття нового формату подання інформації та здатність працювати з QR.
- Розуміння та готовність до співпраці серед викладачів, що є невід’ємною частиною роботи порталу та забезпечення оновлення тестової бази.

На другому етапі отримано та оброблено результати на основі отриманих даних , проілюстровано отримані результати в діаграмах та таблицях , розроблено алгоритми оцінки та накопичення балів для реалізації комплексних винагороджень на певних етапах проходження тестів. Сформовано роботу алгоритма по створенню QR кодів

На третьому етапі створення робочої програми та розгортання її в мережі інтернет.

На четвертому етапі описано все програмне забезпечення та програми, що використовувалися під час розробки додатку. і

Підсумовуючи все вищезазначене, видно ,що додаток надає змогу наблизити формування та проходження тестів до максимально незалежних умов . Кожен тест формується та впроваджується для кожного учасника тестування по алгоритму ,що виключає можливість співпадіння питань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dependency Injection in .NET, Mark Seemann
2. Beginning SOLID Principles and Design Patterns for [ASP.NET](#) Developers, Bipin Joshi
3. Azure for Architects: Implementing Cloud Design, DevOps, IoT, and Serverless Solutions on Your Public Cloud, Ritesh Modi
4. Programming Entity Framework: Code First, Julia Lerman , Rowan Miller
5. Pof EAA: Repository - Martin Fowler
6. 3-Tier Architecture in [ASP.NET](#) with C# tutorial, Donald Wolfe
7. Mastering Microsoft Azure Infrastructure Services, John Savill
8. Cloud Architecture Patterns: Using Microsoft Azure, Bill Wilder
9. CLR VIA C# Jeffrey Richter
10. C# in Depth, Jon Skeet
11. Hands-On Azure for Developers: Implement Rich Azure PaaS Ecosystems Using Containers, Serverless Services, and Storage Solutions, Kamil
12. Design Patterns: Elements of Reusable Object-Oriented Software Grady Booch
13. Head First Design Patterns, Eric Freeman , Elisabeth Robson
14. Patterns of Enterprise Application Architecture , Martin Fowler
15. Pro [ASP.NET](#) Core MVC 2, Adam Freeman
16. Modern API Design with [ASP.NET](#) Core 2: Building Cross-Platform Back-End Systems, Fanie Reynders
17. Programming Entity Framework: Code First, Julia Lerman , Rowan Miller
18. Learning SQL , Alan Beaulieu
19. Logical Database Design Principles, John Garmany, Jeff Walker, Terry Clark Refactoring , Improving the Design of Existing Code , Martin Fowler
20. Applied [Microsoft.NET](#) Framework Programming, Jeffrey Richter
Windows Runtime Via C# , Jeffrey Richter